


```
FFFFFFFFF 000000 RRRRRRRR LL      IIIII 88888888
FFFFFFFFF 000000 RRRRRRRR LL      IIIII 88888888
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FFFFFFFFF 00      00 RRRRRRRR LL      II      88888888
FFFFFFFFF 00      00 RRRRRRRR LL      II      88888888
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        00      00 RR      RR LL      II      88      88
FF        000000 RR      RR LLLLLLLLL IIIII 88888888
FF        000000 RR      RR LLLLLLLLL IIIII 88888888
```

```
LL      IIIII SSSSSSSS
LL      IIIII SSSSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SSSSSS
LL      II     SSSSSS
LL      II     SS
LL      II     SS
LL      II     SS
LL      II     SS
LLLLLLLLL IIIII SSSSSSSS
LLLLLLLLL IIIII SSSSSSSS
```


N 9
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 1
(1)

FORTRAN Run-Time Library specific macros and symbols
File: FORLIB.REQ, Edit: SBL1003

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

Author: Steven B. Lionel, 23-September-1982

1-001 - Original. SBL 23-September-1982
1-002 - Add FORMACROS.REQ. 1-Mar-1983
1-003 - Add FORRCE.R32. SBL 2-Jun-1983

+ This file is the master source for FORLIB.L32.
It contains definitions for macros and symbols used internally to the
FORTRAN Run-Time Library.

SWITCHES ADDRESSING_MODE (EXTERNAL=GENERAL, NONEXTERNAL=WORD_RELATIVE);
LIBRARY 'RTLSTARLE'; ! SYS\$LIBRARY:STARLET.L32
REQUIRE 'RTLML:FORERR'; ! FOR\$K_ error codes

B 10
15-Sep-1984 23:44:38
15-Sep-1984 22:45:53

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.OBJ]FORERR.R32;1

Page 2
(1)

```

: R0045 0
: R0046 0
: R0047 0
: R0048 0
: R0049 0
: R0050 0
: R0051 0
: R0052 0
: R0053 0
: R0054 0
: R0055 0
: R0056 0
: R0057 0
: R0058 0
: R0059 0
: R0060 0
: R0061 0
: R0062 0
: R0063 0
: R0064 0
: R0065 0
: R0066 0
: R0067 0
: R0068 0
: R0069 0
: R0070 0
: R0071 0
: R0072 0
: R0073 0
: R0074 0
: R0075 0
: R0076 0
: R0077 0
: R0078 0
: R0079 0
: R0080 0
: R0081 0
: R0082 0
: R0083 0
: R0084 0
: R0085 0
: R0086 0
: R0087 0
: R0088 0
: R0089 0
: R0090 0
: R0091 0
: R0092 0
: R0093 0
: R0094 0
: R0095 0
: R0096 0
: R0097 0
: R0098 0
: R0099 0
: R0100 0
: R0101 0

! *****
! Created 15-SEP-1984 22:45:50 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:44:40 _S255$DUA28:[FORRTL.SRC]FORE
! *****

!*** MODULE $FORERR ***
!literal FOR$K_FAC NO = 24;
!literal FOR$K_NOTFORSPE = 1;
!literal FOR$K_SYNERRNAM = 17;
!literal FOR$K_TOOMANVAL = 18;
!literal FOR$K_INVREFVAR = 19;
!literal FOR$K_REWERR = 20;
!literal FOR$K_DUPFILSPE = 21;
!literal FOR$K_INPRECTOO = 22;
!literal FOR$K_BACERR = 23;
!literal FOR$K_ENDDURREA = 24;
!literal FOR$K_RECNUMOUT = 25;
!literal FOR$K_OPEDEFREQ = 26;
!literal FOR$K_TOOMANREC = 27;
!literal FOR$K_CLOERR = 28;
!literal FOR$K_FILNOTFOU = 29;
!literal FOR$K_OPEFAI = 30;
!literal FOR$K_MIXFILACC = 31;
!literal FOR$K_INVLOGUNI = 32;
!literal FOR$K_ENDFILERR = 33;
!literal FOR$K_UNIALROPE = 34;
!literal FOR$K_SEGRECFOR = 35;
!literal FOR$K_ATTACCNON = 36;
!literal FOR$K_INCRECLEN = 37;
!literal FOR$K_ERRDURWRI = 38;
!literal FOR$K_ERRDURREA = 39;
!literal FOR$K_RECIO OPE = 40;
!literal FOR$K_INSVIRMEM = 41;
!literal FOR$K_NO SUCDEV = 42;
!literal FOR$K_FILNAMSPE = 43;
!literal FOR$K_INRECTYP = 44;
!literal FOR$K_KEYVALERR = 45;
!literal FOR$K_INCOPECLO = 46;
!literal FOR$K_WRIREFIL = 47;
!literal FOR$K_INVARGFOR = 48;
!literal FOR$K_INVKEYSPE = 49;
!literal FOR$K_INCKEYCHG = 50;
!literal FOR$K_INCFILORG = 51;
!literal FOR$K_SPERECLOC = 52;
!literal FOR$K_NO CURREC = 53;
!literal FOR$K_REWRITERR = 54;
!literal FOR$K_DELERR = 55;
!literal FOR$K_UNLERR = 56;
!literal FOR$K_FINERR = 57;
!literal FOR$K_LISIO SYN = 59;
!literal FOR$K_INFFORLOO = 60;
!literal FOR$K_FORVARMIS = 61;
!literal FOR$K_SYNERRFOR = 62;
!literal FOR$K_OUTCONERR = 63;
!literal FOR$K_INPCONERR = 64;
!literal FOR$K_OUTSTAOVE = 66;
!literal FOR$K_INPSTAREQ = 67;
```


C 10
15-Sep-1984 23:44:38
15-Sep-1984 22:45:53

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.OBJ]FORERR.R32;1 Page 3
(1)

```
: R0102 0 literal FOR$K_VFEVALERR = 68;  
: R0103 0 literal FOR$K_INTOVF = 70;  
: R0104 0 literal FOR$K_INTZERDIV = 71;  
: R0105 0 literal FOR$K_FLOOVE = 72;  
: R0106 0 literal FOR$K_FLOZERDIV = 73;  
: R0107 0 literal FOR$K_FLOUND = 74;  
: R0108 0 literal FOR$K_DECSTROVE = 76;  
: R0109 0 literal FOR$K_ARRREFOUT = 77;  
: R0110 0 literal FOR$K_ADJARRDIM = 93;  
: R0111 0 literal FOR$K_MAX_ERR = 93;
```

D 10
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 4
(1)

: 0112 0
: 0113 0

REQUIRE 'RTLIN:FORFMT';

! FORMAT codes and fields

E 10
15-Sep-1984 23:44:38
15-Sep-1984 22:44:47

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORFMT.REQ;1 Page 5
(1)

File: FORFMT.REQ Edit: JAW1004

* This file, FORFMT.REQ, defines symbols for the VAX-11 FORTRAN
formatting routines.

* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

Revision History:

0-12 - Change name to FORFMT.REQ JBS 14-NOV-78
1-001 - Add copyright notice and change version number JBS 16-NOV-78
1-002 - Add FORTRAN-77 format codes. SBL 09-Feb-1979
1-003 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
1-004 - Add V_RC_TYPE_BYTE and V_RC_TYPE_WORD. JAW 10-Aug-1981
--

* Define format code byte

MACRO

V_FMT_CODE = 0,0,7,0 % ! 7-bit format code
V_FMT_REPRE = 0,7,1,0 % ! Optional representation byte present?

MACRO

* Optional format representation byte:
The following are bits in the optional format representation
byte. The byte is copied into local BLOCK
B_FMT_REPRESENT. These flags indicate less
frequently used sizes of the format code representations.
--

```
V_RC_TYPE = 0,0,2,0 %, Repeat count type:
                          0 = not present, 1 = byte, 2 = word
V_RC_TYPE_BYTE = 0,0,1,0 %,
V_RC_TYPE_WORD = 0,1,1,0 %,
V_W_WORD = 0,2,1,0 %, 0=W field is byte, 1=W field is word.
                          Bit 3 is reserved to DEC
V_E_VFE = 0,4,1,0 %, E field is VFE
V_D_VFE = 0,5,1,0 %, D field is VFE
V_W_VFE = 0,6,1,0 %, W field is a VFE (ignore V_W_WORD)
V_RC_VFE = 0,7,1,0 %: Repeat count field is a VFE
                          ignore V_RC_BYTE and V_RC_WORD)
```

+ Define format code symbols which are is 2 or 3 characters so it plus comma will fit
in between logical tabs. One character symbols are prefixed with so
that they are two character symbols instead (so won't conflict with LOCALs).

LITERAL				
ER	= 0,	00	:	Format syntax error - only from object time format compiler
LP	= 1,	01	:	(- Format reversion point
NLP	= 2,	02	:	n(- Left paren of repeat group
RP	= 3,	03	:) - Right paren of repeat group
EOF	= 4,	04	:) - End of format
SLS	= 5,	05	:	/ - Record separator
DLR	= 6,	06	:	\$ - Dollar sign: terminal I/O
			:	do not return to left margin
CLN	= 7,	07	:	: - Colon: terminate if end of list
			:	so no trailing Hollerith printed
S	= 9,	09	:	S - Restore + optional
SP	= 10,	0A	:	SP - Force + on
SS	= 11,	0B	:	SS - Force + off
P	= 12,	0C	:	sP - signed scale factor (-128 =< s =< +127).
T	= 13,	0D	:	Tn - Tab Set (0 < n =< 32767)
X	= 14,	0E	:	nX - Skip n columns (0 < n =< 32767)
H	= 15,	0F	:	nHcccc - Hollerith: n chars follow (0 < n =< 32767)
BN	= 16,	10	:	BN - Blanks are nulls
BZ	= 17,	11	:	BZ - Blanks are zeroes
TL	= 18,	12	:	TLc - Tab left c columns
TR	= 19,	13	:	TRc - Tab right c columns
Q	= 20,	14	:	Q - no. of input chars left in record
A	= 21,	15	:	nAw - Alpha numeric
	_MIN_DATA = A,		:	Minimum I/O list transmitting data code
L	= 22,	16	:	nLw - Locial
	_MIN_INT = L,		:	Min. integer
O	= 23,	17	:	nOw - Octal
I	= 24,	18	:	nIw - Integer
Z	= 25,	19	:	nZw - Hexadecimal
XO	= 26,	1A	:	Ow.m - Extended 0
XI	= 27,	1B	:	Iw.m - Extended I
XZ	= 28,	1C	:	Zw.m - Extended Z
	_MAX_INT = XZ,		:	Max. integer (not counting defaults)

R0228 0
R0229 0
R0230 0
R0231 0
R0232 0
R0233 0
R0234 0
R0235 0
R0236 0
R0237 0
R0238 0
R0239 0
R0240 0
R0241 0
R0242 0
R0243 0
R0244 0
R0245 0
R0246 0
R0247 0
R0248 0
R0249 0
R0250 0
R0251 0
R0252 0
R0253 0
R0254 0
R0255 0
R0256 0
R0257 0
R0258 0
R0259 0
R0260 0
R0261 0
R0262 0

```

-F      = 30,      1E
      _MIN_FLT = -F, 1F
-E      = 31,      20
-G      = 32,      21
-D      = 33,      22
XE      = 34,      23
XG      = 35,
      MAX_FLT = XG,
      MAX_DATA = XG,

```

+ Default format codes:
-

```

-DA      = 41,      29
-DL      = 42,      2A
-DO      = 43,      2B
-DI      = 44,      2C
-DZ      = 45,      2D

-DF      = 50,      32
-DE      = 51,      33
-DG      = 52,      34
-DD      = 53,      35

```

```

nFw.d - Fixed format
Min. floating
nEw.d - Scientific notation format
nGw.d - General format
nDw.d - Double Precision format
nEw.dEe - Extended E
nGw.dEe - Extended G
max. floating (not counting default)
Max. data (not counting default)

```

```

nA - default A
nL - default L
nO - default O
nI - default I
nZ - default Z

nF - default F
nE - default E
nG - default G
nD - default D

```

+ Note: 0 < n =< 32767 (decimal)
0 < w =< 65535 (decimal)
0 =< d =< 255 (decimal)
0 =< e =< 255 (decimal)
-

! End of file FORFMT.REQ

M 10
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 8 (1)

: 0263 0
: 0264 0

REQUIRE 'RTLIN:FORMACROS';

! FORTRAN-specific macros

Macros for FORTRAN Run-Time Library
File: FORMACROS.REQ, Edit: SBL1001

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

Author: Steven B. Lionel, 7-Jan-1983

1-001 - Original. SBL 7-Jan-1983

+ Macro to call FOR\$\$SIGNAL_STO and return. This saves lines of code
in the source. It can replace a call to FOR\$\$SIGNAL_STO anywhere.

MACRO

```
$FOR$$SIGNAL_STO (signame) =
  BEGIN
    FOR$$SIGNAL_STO (signame
      %IF %NULL(%REMAINING)
      %THEN
      %ELSE
      ,%REMAINING
      %FI
    );
  RETURN 0;
  END %;
```

+ Structure definitions used to declare the FAB and NAM as being offset from
CCB. To use, make the following declarations:

```
BIND
  FAB = CCB: REF $FOR$FAB_CCB_STRUCT,
  NAM = CCB: REF $FOR$NAM_CCB_STRUCT;
```

J 10
15-Sep-1984 23:44:38
15-Sep-1984 22:45:03

VAX-11 Bliss-32 V4.0-742
[FORRTL.SRC]FORMACROS.REQ;1

Page 10
(1)

```

R0322 0
R0323 0
R0324 0
R0325 0
R0326 0
R0327 0
R0328 0
R0329 0
R0330 0
R0331 0
R0332 0
R0333 0
R0334 0
!
!-
STRUCTURE
  $FOR$FAB_CCB_STRUCT [O, P, S, E] =
    [FAB$C_BLN]
    ($FOR$FAB_CCB_STRUCT+RAB$C_BLN+O)<P,S,E>,
  $FOR$NAM_CCB_STRUCT [O, P, S, E] =
    [NAM$C_BLN]
    ($FOR$NAM_CCB_STRUCT+RAB$C_BLN+FAB$C_BLN+O)<P,S,E>;
! End of file FORMACROS.REQ
```


K 10
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_\$255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 11
(1)

: 0335 0
: 0336 0

REQUIRE 'RTLML:FORMSG';

! FOR\$_error codes

R0337 0
R0338 0
R0339 0
R0340 0
R0341 0
R0342 0
R0343 0
R0344 0
R0345 0
R0346 0
R0347 0
R0348 0
R0349 0
R0350 0
R0351 0
R0352 0
R0353 0
R0354 0
R0355 0
R0356 0
R0357 0
R0358 0
R0359 0
R0360 0
R0361 0
R0362 0
R0363 0
R0364 0
R0365 0
R0366 0
R0367 0
R0368 0
R0369 0
R0370 0
R0371 0
R0372 0
R0373 0
R0374 0
R0375 0
R0376 0
R0377 0
R0378 0
R0379 0
R0380 0
R0381 0
R0382 0
R0383 0
R0384 0
R0385 0
R0386 0
R0387 0
R0388 0
R0389 0
R0390 0
R0391 0
R0392 0
R0393 0

Created 15-SEP-1984 22:46:32 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:45:39 _\$255\$DUA28:[FORRTL.OBJ]FORM

*** MODULE \$FORDEF ***

This SDL File Generated by VAX-11 Message V04-00 on 15-SEP-1984 22:45:40.89
FILE: FORMSG.MSG EDIT: SBL2006

*
* COPYRIGHT (C) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*

- 1-014 - ADD ERRORS 17-19 FOR NAMELIST. SBL 10-NOV-1980
- 1-015 - CORRECT TYPO IN TOOMANVAL. SBL 15-DEC-1980
- 1-016 - RESTORE CLOSING BRACKET ON OPEDEFREQ, MYSTERIOUSLY DROPPED IN 1-015.
SBL 6-JAN-1981
- 2-001 - CONVERT TO MESSAGE UTILITY SOURCE FORMAT. ADAPTED FROM FORMSG.MDL
VERSION 1-016. SBL 22-APR-1981
- 2-002 - ADD UNFIO_FMT, FMTIO_UNF, DIRIO_KEY, SEQIO_DIR, KEYIO_DIR,
IO_NONFOR, INVXTREC, FLOUNDEXC. JAW 23-AUG-1981
- 2-003 - CHANGE INVXTREC TO INVTEXREC. JAW 24-AUG-1981
- 2-004 - ADD INVTEX WHICH DOES NOT REQUIRE A RECORD NUMBER. THIS IS FOR INDEXED
AND INTERNAL FILES THAT DO NOT HAVE A VALID RECORD NUMBER. DGP 21-DEC-9181
- 2-005 - ADD .TITLE. SBL 28-AUG-1982
- 2-006 - ADD OPEREQDIS AND OPEREQSEQ. SBL 2-JUN-1983

SYMBOLS ARE DEFINED FOLLOWING THE STANDARD FOR GLOBAL NAMES:

FOR\$ _ABCMNOXYZ

IN ADDITION, THE LIBRARY STANDARDS SPECIFY THAT THE LETTERS "abc", "mno",
AND "xyz" ARE THE FIRST THREE LETTERS OF THE FIRST THREE WORDS OF THE ERROR
MESSAGE, NOT COUNTING ARTICLES AND PREPOSITIONS.

R0394 0
R0395 0
R0396 0
R0397 0
R0398 0
R0399 0
R0400 0
R0401 0
R0402 0
R0403 0
R0404 0
R0405 0
R0406 0
R0407 0
R0408 0
R0409 0
R0410 0
R0411 0
R0412 0
R0413 0
R0414 0
R0415 0
R0416 0
R0417 0
R0418 0
R0419 0
R0420 0
R0421 0
R0422 0
R0423 0
R0424 0
R0425 0
R0426 0
R0427 0
R0428 0
R0429 0
R0430 0
R0431 0
R0432 0
R0433 0
R0434 0
R0435 0
R0436 0
R0437 0
R0438 0
R0439 0
R0440 0
R0441 0
R0442 0
R0443 0
R0444 0
R0445 0
R0446 0
R0447 0
R0448 0
R0449 0
R0450 0

THE NAMES OF ALL PROCEDURES USING EACH ERROR CONDITION VALUE SHOULD APPEAR
IN THE COMMENTS INCLUDED WITH EACH ERROR DEFINITION.

MACRO-32 PROGRAMMING:

THE MACROS CALL:

\$FORDEF

WILL CAUSE ALL SYMBOLS TO BE MADE AVAILABLE TO THE MODULE.
THE STSSV MSG_ID IS THE FORTRAN ERROR NUMBER (1:94).
THESE SYMBOLS ARE DECLARED EXTERNAL BY THE RUN-TIME LIBRARY,
THE MACROS ARE PROVIDED ONLY FOR THE CONVENIENCE OF THE USER.

```

; MAKE ALL ERRORS BE SEVERE (EXCEPT AS NOTED)
; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
; SET LH TO 24 (DECIMAL).
literal FOR$_FACILITY = 24;
literal FOR$_NOTFORSPE = 1605644;
; NEVER SIGNALLED. USED ONLY AS A FORTRAN ERROR
; NUMBER FOR ERRSNS TO MEAN SOME OTHER FACILITY
; THAN FOR$ OR MTH$ DETECTED THE ERROR.
; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
; SET LH TO 24 (DECIMAL).
literal FOR$_SYNERRNAM = 1605772;
literal FOR$_TOOMANVAL = 1605780;
literal FOR$_INVREFVAR = 1605788;
literal FOR$_REWERR = 1605796;
literal FOR$_DUPFILSPE = 1605804;
literal FOR$_INPRECTOO = 1605812;
literal FOR$_BACERR = 1605820;
literal FOR$_ENDDURREA = 1605828;
literal FOR$_RECNUMOUT = 1605836;
literal FOR$_OPEDEFREQ = 1605844;
literal FOR$_TOOMANREC = 1605852;
literal FOR$_CLOERR = 1605860;
literal FOR$_FILNOTFOU = 1605868;
literal FOR$_OPEFAI = 1605876;
literal FOR$_MIXFILACC = 1605884;
literal FOR$_INVLOGUNI = 1605892;
literal FOR$_ENDFILERR = 1605900;
literal FOR$_UNIALROPE = 1605908;
literal FOR$_SEGRECFOR = 1605916;
literal FOR$_ATTACCNON = 1605924;
literal FOR$_INCRECLEN = 1605932;
literal FOR$_ERRDURWRI = 1605940;
literal FOR$_ERRDURREA = 1605948;
literal FOR$_RECIO OPE = 1605956;
literal FOR$_INSVIRMEM = 1605964;
literal FOR$_NO SUCDEV = 1605972;
literal FOR$_FILNAMSPE = 1605980;
literal FOR$_INRECTYP = 1605988;
literal FOR$_KEYVALERR = 1605996;
literal FOR$_INCOPECLO = 1606004;
literal FOR$_WRIREFIL = 1606012;
literal FOR$_INVARGFOR = 1606020;

```

N 10
15-Sep-1984 23:44:38
15-Sep-1984 22:46:38

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.OBJ]FORMSG.R32;1 Page 14
(1)

```
R0451 0 literal FOR$ INVKEYSPE = 1606028;
R0452 0 literal FOR$ INCKEYCHG = 1606036;
R0453 0 literal FOR$ INCFILORG = 1606044;
R0454 0 literal FOR$ SPERECLOC = 1606052;
R0455 0 literal FOR$ NO CURREC = 1606060;
R0456 0 literal FOR$ REWRITERR = 1606068;
R0457 0 literal FOR$ DELERR = 1606076;
R0458 0 literal FOR$ UNLERR = 1606084;
R0459 0 literal FOR$ FINERR = 1606092;
R0460 0 literal FOR$ MORONEREC = 1605852;
R0461 0 literal FOR$ ATTREANON = 1605924;
R0462 0 ; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
R0463 0 literal FOR$ LISIO SYN = 1606108;
R0464 0 literal FOR$ INFFORLOO = 1606116;
R0465 0 literal FOR$ FORVARMIS = 1606124;
R0466 0 literal FOR$ SYNERRFOR = 1606132;
R0467 0 ; SET SUB-SYSTEM SPECIFIC BIT AND MAKE ERROR (NOT SEVERE)
R0468 0 literal FOR$ OUTCONERR = 1606138;
R0469 0 ; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
R0470 0 literal FOR$ INPCONERR = 1606148;
R0471 0 ; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
R0472 0 literal FOR$ OUTSTAOVE = 1606164;
R0473 0 literal FOR$ INPSTAREQ = 1606172;
R0474 0 literal FOR$ VFEVALERR = 1606180;
R0475 0 ; SET SUB-SYSTEM SPECIFIC BIT AND MAKE SEVERE
R0476 0 literal FOR$ ADJARRDIM = 1606380;
R0477 0 *****
R0478 0 THE FOLLOWING MESSAGES ARE SECONDARY MESSAGES, OR ARE USED ONLY IN
R0479 0 EXIT HANDLERS, AND THEREFORE DO NOT NEED TO BE (AND SHOULD NOT BE)
R0480 0 CONTIGUOUS WITH THE MESSAGES ABOVE.
R0481 0 THE NEXT SIX MESSAGES ARE SECONDARY MESSAGES FOR USE WITH MIXFILACC IN
R0482 0 FOR$$IO BEG AND FOR$$CB.
R0483 0 literal FOR$ UNFIO_FMT = 1607684;
R0484 0 literal FOR$_FMTIO_UNF = 1607692;
R0485 0 literal FOR$ DIRIO_KEY = 1607700;
R0486 0 literal FOR$ SEQIO_DIR = 1607708;
R0487 0 literal FOR$ KEYIO_DIR = 1607716;
R0488 0 literal FOR$ IO_NONFOR = 1607724;
R0489 0 THE NEXT TWO MESSAGES ARE SECONDARY MESSAGES FOR USE WITH INPCONERR IN
R0490 0 FOR$$UDF WF AND FOR$$UDF WL.
R0491 0 literal FOR$ INVTEXREC = 1607732;
R0492 0 literal FOR$ INVTEX = 1607740;
R0493 0 ADDITIONAL SECONDARY MESSAGES
R0494 0 literal FOR$ OPEREQDIS = 1607748;
R0495 0 literal FOR$ OPEREQSEQ = 1607756;
R0496 0 THE NEXT MESSAGE IS A PRIMARY MESSAGE USED IN THE EXIT HANDLER WHICH
R0497 0 IS DECLARED BY FOR$HANDLER.
R0498 0 literal FOR$ FLOUNDEXC = 1608035;
R0499 0 END OF SPECIAL MESSAGES
R0500 0 *****
```


B 11
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_ \$255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 15
(1)

: 0501 0
: 0502 0

REQUIRE 'RTLIN:FORMML';

! NAMELIST definitions

:
:

! FORMML.REQ - NAMELIST NML\$ Definitions - Version 1-002 - Edit: SBL1002

```

*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****

```

! AUTHOR: Steven B. Lionel

! EDIT HISTORY:

! 1-001 - Original. SBL 21-August-1980
 ! 1-002 - Add NML\$V_SUBSCRIPT. SBL 15-April-1981

LITERAL

NML\$K_BLKLENGTH = TPASK_LENGTH0 + 120; ! TPARSE parameter block length

FIELD

NML\$FIELDS =

+ The following fields are part of the TPARSE parameter block past that
 - used by LIB\$TPARSE.

SET

NML\$A_LISTBLOCK	= [TPASK_LENGTH0 + 00, 0, 32, 0],	! NAMELIST descriptor block
NML\$A_VARNAME	= [TPASK_LENGTH0 + 04, 0, 32, 0],	! Variable name
NML\$A_VARSTART	= [TPASK_LENGTH0 + 08, 0, 32, 0],	! Variable start address
NML\$A_VAREND	= [TPASK_LENGTH0 + 12, 0, 32, 0],	! Variable end address
NML\$A_VARCUR	= [TPASK_LENGTH0 + 16, 0, 32, 0],	! Current position
NML\$W_VARSIZE	= [TPASK_LENGTH0 + 20, 0, 16, 0],	! Segment size
NML\$W_STRIDE	= [TPASK_LENGTH0 + 22, 0, 16, 0],	! Stride between elements
NML\$A_DESCR	= [TPASK_LENGTH0 + 24, 0, 32, 0],	! Address of descriptor
NML\$A_CCB	= [TPASK_LENGTH0 + 28, 0, 32, 0],	! Address of CCB
NML\$B_DTYPE	= [TPASK_LENGTH0 + 32, 0, 08, 0],	! Variable datatype
NML\$V_SUBSTRING	= [TPASK_LENGTH0 + 33, 0, 01, 0],	! Set if substring
NML\$V_IMAG	= [TPASK_LENGTH0 + 33, 1, 01, 0],	! Set if imaginary part

D 11
15-Sep-1984 23:44:38
15-Sep-1984 22:45:13

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORNML.REQ;1 Page 17
(1)

```
: R0560 0      NML$V_VALUE_IDENT =
: R0561 0      NML$V_SUBSCRIPT = [TPASK_LENGTH0 + 33, 2, 01, 0],
: R0562 0      NML$B_CONSTYPE = [TPASK_LENGTH0 + 33, 3, 01, 0],
: R0563 0      NML$L_CURIDX = [TPASK_LENGTH0 + 34, 0, 08, 0],
: R0564 0      NML$L_SUBSCR = [TPASK_LENGTH0 + 36, 0, 32, 0],
: R0565 0      NML$L_SUBSTRLO = [TPASK_LENGTH0 + 40, 0, 32, 0],
: R0566 0      NML$L_SUBSTRHI = [TPASK_LENGTH0 + 40, 0, 32, 0],
: R0567 0      NML$L_CONSBLOCK = [TPASK_LENGTH0 + 44, 0, 32, 0],
: R0568 0      NML$L_REPEATCT = [TPASK_LENGTH0 + 68, 0, 32, 0],
: R0569 0      NML$T_TOKEN = [TPASK_LENGTH0 + 84, 0, 32, 0],
: R0570 0
: R0571 0
: R0572 0
: R0573 0
: R0574 0
: R0575 0      TES;
: R0576 0      ! End of FORNML.REQ
: R0577 0
```

```
: Set if last token was an identifier
: Set of a subscript has been seen
: Constant type
: Current index number
: Subscripts (7 longwords)
: Low substring column
: High substring column
: Constant storage block
: 4 longwords
: Repeat count
: Last token if
: it could be an
: identifier.
: 32 bytes long.
```

E 11
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 18
(1)

: 0578 0
: 0579 0

REQUIRE 'RTLIN:FOROPN';

! OPEN definitions

* This file, FOROPN.REQ, defines the VAX-11 FORTRAN OPEN,
CLOSE and INQUIRE keywords and literal values. Edit: SBL1023

* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

- *****
0-21 - Move parameter encoding symbols for FOR\$\$IOBEG to FPAR. TNH 30-May-78
0-22 - And symbol for statement types. TNH 30-May-78
0-23 - Change name to FOROPN.REQ JBS 14-NOV-78
1-001 - Increment version number and add copyright notice JBS 16-NOV-78
1-002 - Add Some symbolics to define the special LUN numbers for Basic
PRINT, INPUT, READ. DGP 05-Dec-78
1-003 - Add necessary symbolics for ISAM. SBL 06-Dec-78
1-004 - Change file name from FOROPN.REQ to OTSOPN.REQ. JBS 06-DEC-78
1-005 - Remove the statement type constants. Moved to LUB. DGP 06-Dec-78
1-006 - Change back to FOROPN. Move more constants to LUB. DGP 08-Dec-78
1-007 - Fix some comments to reflect the change back to FOROPN. JBS 12-DEC-78
1-008 - Add DISP='SUBMIT'. SBL 09-Feb-1979
1-009 - Change ORG_IND to ORG_IDX. Add new arg type. SBL 03-Apr-79
1-010 - Add new definitions for ISAM. SBL 6-Apr-79
1-011 - Give BLANK= literal values. SBL 12-Apr-79
1-012 - Add OPEN\$K_DIS SUDE and OPEN\$K_DIS PRDE. SBL 19-Apr-79
1-013 - Add OPEN\$K_ISTAT_L. SBL 27-Apr-79
1-014 - Add INQUIRE keywords. SBL 27-Apr-79
1-015 - Because OPEN and CLOSE use the IOSTAT keywords, make the
symbols that control the size of their keyword arrays
cover them. JBS 01-MAY-1979
1-016 - Reassign INQUIRE keywords. SBL 01-May-1979
1-017 - Add INQUIRE keyword values for ORGANIZATION, RECORDTYPE
and KEYED. SBL 2-Aug-1979
1-018 - Add OPEN\$K_ARG B R. SBL 7-August-1979
1-019 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
1-020 - Add CARRIAGECONTROL for INQUIRE. SBL 4-Dec-1979

- 1-021 - ISAM KEY positions in the OPEN arg list are signed longwords,
not words. The key length should be an unsigned byte. SBL 12-Mar-1980
1-022 - Add OPEN\$K_DEFAULT and INQ\$K_DEFAULT. JAW 30-Jun-1981
1-023 - Add STREAM, STREAM_CR and STREAM_LF values for RECORDTYPE. SBL 1-Mar-1983

--
+
Define symbols for FORTRAN OPEN keywords of form: OPEN\$K_symbol
Define literal values of form: OPEN\$K_abc_xyz where abc is
first three letters of keyword and xyz are the first three
letters of the literal.
Define symbols in alphabetical order.

LITERAL

OPEN\$K_ACCESS = 4,
OPEN\$K_ACC_DIR = 1,
OPEN\$K_ACC_SEQ = 2,
OPEN\$K_ACC_APP = 3,
OPEN\$K_ACC_KEY = 4,

OPEN\$K_ASSOCIAT = 17,
OPEN\$K_ASSOC_L = 0,

OPEN\$K_BLANK = 24,
OPEN\$K_BLK_ZER = 1,
OPEN\$K_BLK_NUL = 2,

OPEN\$K_BLOCKSIZE = 18,
OPEN\$K_BUFFERCO = 9,
OPEN\$K_CARRIAGE = 7,

OPEN\$K_CAR_FOR = 1,
OPEN\$K_CAR_LIS = 2,
OPEN\$K_CAR_NON = 3,

OPEN\$K_DEFAULTFILE = 26,
OPEN\$K_DISPOSE = 2,
OPEN\$K_DIS_SAV = 1,

OPEN\$K_DIS_DEL = 2,
OPEN\$K_DIS_PRI = 3,
OPEN\$K_DIS_SUB = 4,

OPEN\$K_DIS_PRDE = 5,
OPEN\$K_DIS_SUDE = 6,
OPEN\$K_ERR = 3,

OPEN\$K_EXTENDSI = 11,
OPEN\$K_FORM = 5,
OPEN\$K_FOR_FOR = 1,

OPEN\$K_FOR_UNF = 2,
OPEN\$K_FOR_UN = -1,

OPEN\$K_INITIALS = 10,
OPEN\$K_ISTAT = 22,
OPEN\$K_ISTAT_L = 25,

OPEN\$K_KEY = 23,

ACCESS
= 'DIRECT'
= 'SEQUENTIAL'
= 'APPEND'
= 'KEYED'

ASSOCIATEDVARIABLE

1 if associated variable is a longword
0 if just a word. Note: this parameter
is not generated by the compiler!
It is needed after all keywords are converted to
32-bit values.

BLANK
= 'ZERO'
= 'NULL'
BLOCKSIZE
BUFFERCOUNT
CARRIAGE CONTROL

= 'FORTRAN'
= 'LIST'
= 'NONE'

DEFAULTFILE
DISPOSE
= 'SAVE'

= 'DELETE'
= 'PRINT'
= 'SUBMIT'

= 'PRINT/DELETE'
= 'SUBMIT/DELETE'

ERR
EXTENDSI
! FORM

= 'FORMATTED'
= 'UNFORMATTED'
= 'UNSPECIFIED'

Note: this is not generated by compiler.
It is used by default OPEN only.

INITIALSIZE
ISTAT

1 If IOSTAT is a longword,
0 if a word. This is not generated
by the compiler.

KEY


```

R0694 0
R0695 0
R0696 0
R0697 0
R0698 0
R0699 0
R0700 0
R0701 0
R0702 0
R0703 0
R0704 0
R0705 0
R0706 0
R0707 0
R0708 0
R0709 0
R0710 0
R0711 0
R0712 0
R0713 0
R0714 0
R0715 0
R0716 0
R0717 0
R0718 0
R0719 0
R0720 0
R0721 0
R0722 0
R0723 0
R0724 0
R0725 0
R0726 0
R0727 0
R0728 0
R0729 0
R0730 0
R0731 0
R0732 0
R0733 0
R0734 0
R0735 0
R0736 0
R0737 0
R0738 0
R0739 0
R0740 0
R0741 0
R0742 0
R0743 0
R0744 0
R0745 0
R0746 0
R0747 0
R0748 0
R0749 0
R0750 0

```

```

OPEN$K_MAXREC = 16,
OPEN$K_NAME = 14,
OPEN$K_NOSPANBL = 12,
OPEN$K_USEROPEN = 21,
OPEN$K_ORGANIZA = 19,
    OPEN$K_ORG_SEQ = 1,
    OPEN$K_ORG_REL = 2,
    OPEN$K_ORG_IDX = 3,
    OPEN$K_ORG_HAS = 4,
    OPEN$K_ORG_STR = 5,
OPEN$K_READONLY = 8,
OPEN$K_RECORDTY = 20,
    OPEN$K_REC_FIX = 1,
    OPEN$K_REC_VAR = 2,
    OPEN$K_REC_SEGM = 3,
    OPEN$K_REC_STM = 4,
    OPEN$K_REC_STMCR = 5,
    OPEN$K_REC_STMLF = 6,
OPEN$K_RECORDSI = 6,
OPEN$K_SHARED = 13,
OPEN$K_TYPE = 15,
    OPEN$K_TYP_OLD = 1,
    OPEN$K_TYP_NEW = 2,
    OPEN$K_TYP_SCR = 3,
    OPEN$K_TYP_UNK = 4,
OPEN$K_UNIT = T,

```

```

MAXREC
NAME
NOSPANBLOCKS
USEROPEN
ORGANIZATION
    = 'SEQUENTIAL'
    = 'RELATIVE'
    = 'INDEXED'
    = 'HASHED'
    = 'STREAM'
READONLY
RECORDTYPE
    = 'FIXED'
    = 'VARIABLE'
    = 'SEGMENTED'
    = 'STREAM'
    = 'STREAM_CR'
    = 'STREAM_LF'
RECORDSIZE
SHARED
TYPE
    = 'OLD'
    = 'NEW'
    = 'SCRATCH'
    = 'UNKNOWN'
UNIT

```

```

OPEN$K_KEY_MAX = OPEN$K_DEFAULTF, ! Max. open parameter
CLOSE$K_KEY_MAX = OPEN$K_DEFAULTF; ! Max. CLOSE parameter

```

Key numbers 27-29 are reserved for future OPEN/CLOSE use.

INQUIRE keyword definitions

LITERAL

```

INQ$K_FILE = OPEN$K_NAME,      ! Input file name
INQ$K_DEFAULTF = OPEN$K_DEFAULTF, ! Defaultfile
INQ$K_UNIT = OPEN$K_UNIT,      ! Input unit number
INQ$K_IOSTAT = OPEN$K_IOSTAT,  ! IOSTAT
INQ$K_IOSTAT_L = OPEN$K_IOSTAT_L, ! 1 if IOSTAT is a longword
                                     ! 0 if a word
INQ$K_ERR = OPEN$K_ERR,        ! 1 if ERR= present
INQ$K_EXIST = 30,              ! File exists?
INQ$K_OPENED = 31,             ! File opened?
INQ$K_NUMBER = 32,             ! Open on what unit?
INQ$K_NAMED = 33,              ! Does it have a name?
INQ$K_NAME = 34,               ! What's its name?
INQ$K_ACCESS = 35,             ! Access mode?
INQ$K_SEQUENTIAL = 36,         ! Is it sequential?
INQ$K_DIRECT = 37,             ! Is it direct?
INQ$K_FORM = 38,               ! What's the form?
INQ$K_FORMATTED = 39,          ! Formatted?
INQ$K_UNFORMAT = 40,           ! Unformatted?

```

```

R0751 0
R0752 0
R0753 0
R0754 0
R0755 0
R0756 0
R0757 0
R0758 0
R0759 0
R0760 0
R0761 0
R0762 0
R0763 0
R0764 0
R0765 0
R0766 0
R0767 0
R0768 0
R0769 0
R0770 0
R0771 0
R0772 0
R0773 0
R0774 0
R0775 0
R0776 0
R0777 0
R0778 0
R0779 0
R0780 0
R0781 0
R0782 0
R0783 0
R0784 0
R0785 0
R0786 0
R0787 0
R0788 0
R0789 0
R0790 0
R0791 0
R0792 0
R0793 0
R0794 0
R0795 0
R0796 0
R0797 0
R0798 0
R0799 0
R0800 0
R0801 0
R0802 0
R0803 0
R0804 0
R0805 0
R0806 0
R0807 0

```

```

INQ$K_RECL = 41,
INQ$K_NEXTREC = 42,
INQ$K_BLANK = 43,
INQ$K_ORGANIZAT = 44,
INQ$K_RECORDTYP = 45,
INQ$K_KEYED = 46,
INQ$K_CARRIAGE = 47,

```

```

! What's the recordsize?
! What's the next record
! What are blanks?
! What's the organization?
! What's the recordtype?
! KEYED allowed?
! What's the carriage control?

```

```
INQ$K_KEY_MAX = INQ$K_CARRIAGE;
```

```

!+
! Define FORTRAN OPEN argument type codes.
! Used in field OPEN$B_ARG_TYPE
!-

```

LITERAL

```

OPEN$K_ARG_NULL = 0, ! keyword with no value
OPEN$K_ARG_LIT = 1, ! literal value in W_INFO_WORD
OPEN$K_ARG_W_V = 2, ! expression in W_INFO_WORD
OPEN$K_ARG_W_R = 3, ! next actual is adr. of word
OPEN$K_ARG_L_V = 4, ! next actual is longword value
OPEN$K_ARG_L_R = 5, ! next actual is adr. of longword value
OPEN$K_ARG_TZ_R = 6, ! next actual is adr. of ASCII string
! (needed for compatibility- descriptor
! is the preferred form)
OPEN$K_ARG_T_DS = 7, ! next actual is adr. of string descriptor
OPEN$K_ARG_ZI = 8, ! next actual is proc. adr.
OPEN$K_ARG_INLN = 9, ! next INFO_WORD longwords are arg.
OPEN$K_ARG_B_R = 10, ! next actual is address of byte

```

```
OPEN$K_ARG_MAX = OPEN$K_ARG_B_R; ! max. arg type code
```

```

!+
! Define fields within FORTRAN OPEN parameters
!-

```

MACRO

```

OPEN$B_KEY = 0,0,8,0 %, ! keyword code. Codes are of form:
OPEN$K_keyname
OPEN$B_ARG_TYPE = 0,8,8,0 %, ! arg type code. Codes are
! of form: OPEN$K_ARG_type
OPEN$W_INFO = 0,16,16,1 %, ! 16-bit information.
! sign extend to 32-bits.
OPEN$A_VALUE = 0,0,%BPADDR,0 %, ! Address of value - in next
! position in parameter list
OPEN$G_VALUE = 0,0,%BPVAL,0 %, ! General value - in next
! position in parameter list

```

```

!+
! Macros and literals for KEY= keyword and for ISAM
!-

```

LITERAL

```
OPEN$K_XAB_SIZE = XAB$C_KEYLEN + 4;
```

MACRO

J 11
15-Sep-1984 23:44:38
15-Sep-1984 22:45:18

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FOROPN.REQ;1 Page 23
(1)

```

: R0808 0      OPENS$_DTYPE = 0,0,8,0%,      ! Key datatype in OPEN list
: R0809 0      OPENS$ _KEY_LO = 4,0,32,1%,    ! Low key position in OPEN list
: R0810 0      OPENS$ _KEY_HI = 8,0,32,1%,    ! High key position in OPEN list
: R0811 0      OPENS$ _KTYPE = XAB$C _KEYLEN,0,8,0%, ! Saved datatype
: R0812 0      OPENS$ _SIZE = XAB$C _KEYLEN,8,8,0%, ! Saved key size
: R0813 0      OPENS$ _POS0 = XAB$C _KEYLEN,16,16,0%, ! Saved low position
: R0814 0
: R0815 0
: R0816 0      !+ Max. length of ASCIIZ string for FORTRAN OPEN file name array
: R0817 0      !-
: R0818 0
: R0819 0      LITERAL
: R0820 0      OPENS$ _STR_MAX = 100; ! Max. length of an ASCIIZ string
: R0821 0      ! (arg type TZ_R only). No limit
: R0822 0      ! for string descriptor strings
: R0823 0
: R0824 0      !+
: R0825 0      ! Constants used in parameter encoding between the I/O statement routines
: R0826 0      ! and routine FOR$$IO_BEG. The codes are
: R0827 0      ! both bit positions in the flag word and an index into
: R0828 0      ! a table used for sorting out the parameters.
: R0829 0      ! All are optional for some I/O statement.
: R0830 0      !-
: R0831 0      LITERAL
: R0832 0      K_UNIT      = 0,      ! user supplied unit number
: R0833 0      K_CHAR_COUNT = 1,      ! size of user supplied record for EN/DECODE
: R0834 0      K_REC_NO    = 2,      ! user supplied record number
: R0835 0      K_FMT_ADR   = 3,      ! user supplied format address
: R0836 0      K_USR_BUF_ADR = 4,      ! user supplied buffer for EN/DECODE
: R0837 0      K_OBJ_TIME_FMT = 7;    ! bit says object time format
: R0838 0
: R0839 0
: R0840 0
: R0841 0      !      End of file FOROPN.REQ
```

K 11
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 24
(1)

: 0842 0
: 0843 0

REQUIRE 'RTLML:FORPAR';

! Assorted definitions

L 11
15-Sep-1984 23:44:38
15-Sep-1984 22:46:18

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.OBJ]FORPAR.R32;1

Page 25
(1)

R0844 0
R0845 0
R0846 0
R0847 0
R0848 0
R0849 0
R0850 0
R0851 0
R0852 0
R0853 0
R0854 0
R0855 0
R0856 0
R0857 0
R0858 0
R0859 0
R0860 0
R0861 0
R0862 0
R0863 0
R0864 0
R0865 0

! *****
! Created 15-SEP-1984 22:46:18 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:45:22 _S255\$DUA28:[FORRTL.SRC]FORP
! *****

!*** MODULE \$FORPAR ***
literal FOR\$K_CLASS_SB = 191;
literal FOR\$K_CLASS_NL = 190;
literal FOR\$K_CONTROL_Z = 26;
literal FOR\$K_UNWINDPOP = 0;
literal FOR\$K_UNWINDNOP = 1;
literal FOR\$K_UNWINDRET = 2;
literal FOR\$\$FOR\$R_PAR = 2;
macro FOR\$r_union_1 = 0,0,16,0 %;
literal FOR\$\$s_union_1 = 2;
macro FOR\$r_structure_1 = 0,0,16,0 %;
literal FOR\$\$s_structure_1 = 2;
macro FOR\$B_STMT_TYPE = 0,0,8,0 %;
macro FOR\$B_STMT_FLAGS = 1,0,8,0 %;
macro FOR\$r_structure_2 = 0,0,16,0 %;
literal FOR\$\$s_structure_2 = 2;
macro FOR\$V_OBJ_FMT = 0,8,1,0 %;

M 11
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 26
(1)

: 0866 0
: 0867 0

REQUIRE 'RTLML:FORRCE';

! RFA Cache Entry structure


```

R0868 0
R0869 0
R0870 0
R0871 0
R0872 0
R0873 0
R0874 0
R0875 0
R0876 0
R0877 0
R0878 0
R0879 0
R0880 0
R0881 0
R0882 0
R0883 0
R0884 0
R0885 0
R0886 0
R0887 0
R0888 0
R0889 0
R0890 0
R0891 0
R0892 0
R0893 0
R0894 0
R0895 0
R0896 0
R0897 0

```

```

*****
Created 15-SEP-1984 22:46:24 by VAX-11 SDL V2.0      Source: 15-SEP-1984 22:45:34 _$255$DUA28:[FORRTL.SRC]FORR
*****

*** MODULE RCEDEF IDENT 1-001 ***
+
An RFA Cache Entry (RCE) contains information about previous records
in the file for use by FOR$BACKSPACE, which implements the FORTRAN
BACKSPACE statement.

For sequential organization and access disk files, each time a new record
is read or written, an entry is added to the RFA cache. The cache itself
is a circularly-linked list, established when the file is opened.

-
literal RCE_K_CACHE SIZE = 20;          ! Number of entries in cache
literal RCE_S_RFA UNION = 8;
FIELD RCE_STRUCT$FIELDSET =
SET
  RCE_A_NEXT = [0,0,32,0] ,              ! Pointer to next entry
  RCE_A_PREV = [4,0,32,0] ,              ! Pointer to previous entry
  RCE_L_LOG_RECNO = [8,0,32,0] ,          ! Logical record number for this entry
  RCE_Q_RFA = [12,0,0,0] ,               ! RFA for this entry
  RCE_L_RFA0 = [12,0,32,0] ,              ! First 4 bytes of RFA
  RCE_W_RFA4 = [16,0,16,0] ,              ! Last 2 bytes of RFA
  RCE_R_RFA_STRUCT = [12,0,0,0] ,
  RCE_R_RFA_UNION = [12,0,0,0]
TES:
literal RCE_S_RCE_STRUCT = 20;
MACRO RCE_R_RCE_STRUCT = BLOCK [RCE_S_RCE_STRUCT,byte] FIELD (RCE_STRUCT$FIELDSET) %;

```

: 0898 0
: 0899 0

REQUIRE 'RTLML:OTSISB.BLF';

! Intra-statement block definitions


```

: R0900 0
: R0901 0
: R0902 0
: R0903 0
: R0904 0
: R0905 0
: R0906 0
: R0907 0
: R0908 0
: R0909 0
: R0910 0
: R0911 0
: R0912 0
: R0913 0
: R0914 0
: R0915 0
: R0916 0
: R0917 0
: R0918 0
: R0919 0
: R0920 0
: R0921 0
: R0922 0
: R0923 0
: R0924 0
: R0925 0
: R0926 0
: R0927 0
: R0928 0
: R0929 0
: R0930 0
: R0931 0
: R0932 0
: R0933 0
: R0934 0
: R0935 0
: R0936 0
: R0937 0
: R0938 0
: R0939 0
: R0940 0
: R0941 0
: R0942 0
: R0943 0
: R0944 0
: R0945 0
: R0946 0
: R0947 0
: R0948 0
: R0949 0
: R0950 0
: R0951 0
: R0952 0
: R0953 0
: R0954 0
: R0955 0
: R0956 0

! *****
! Created 15-SEP-1984 22:48:56 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:46:55 _$255$DUA28:[LIBRTL.SRC]OTSI
! *****

!*** MODULE $ISBDEF ***
literal ISB$K-ST-TY-WSF = 1;
literal ISB$K-FORSTTYLO = 1;
literal ISB$K-ST-TY-RSF = 2;
literal ISB$K-ST-TY-WSU = 3;
literal ISB$K-ST-TY-RSU = 4;
literal ISB$K-ST-TY-WDF = 5;
literal ISB$K-ST-TY-RDF = 6;
literal ISB$K-ST-TY-WDU = 7;
literal ISB$K-ST-TY-RDU = 8;
literal ISB$K-ST-TY-WSL = 9;
literal ISB$K-ST-TY-RSL = 10;
literal ISB$K-ST-TY-WMF = 11;
literal ISB$K-MIN-DE-EN = 11;
literal ISB$K-ST-TY-RMF = 12;
literal ISB$K-MAX-DE-EN = 12;
literal ISB$K-ST-TY-QXF = 13;
literal ISB$K-ST-TY-RKF = 14;
literal ISB$K-ST-TY-WXU = 15;
literal ISB$K-ST-TY-RKU = 16;
literal ISB$K-ST-TY-WIF = 17;
literal ISB$K-ST-TY-RIF = 18;
literal ISB$K-ST-TY-WSN = 19;
literal ISB$K-ST-TY-RSN = 20;
literal ISB$K-ST-TY-WIL = 21;
literal ISB$K-ST-TY-RIL = 22;
literal ISB$K-FORSTTYHI = 22;
literal ISB$K-ST-TY-PRI = 27;
literal ISB$K-BASSTTYLO = 27;
literal ISB$K-ST-TY-LIN = 28;
literal ISB$K-ST-TY-PSE = 29;
literal ISB$K-ST-TY-INP = 30;
literal ISB$K-ST-TY-PRU = 31;
literal ISB$K-ST-TY-INL = 32;
literal ISB$K-ST-TY-DEL = 33;
literal ISB$K-ST-TY-REA = 34;
literal ISB$K-ST-TY-UPD = 35;
literal ISB$K-ST-TY-GSE = 36;
literal ISB$K-ST-TY-RES = 37;
literal ISB$K-ST-TY-SCR = 38;
literal ISB$K-ST-TY-PRE = 39;
literal ISB$K-ST-TY-GRE = 40;
literal ISB$K-ST-TY-FRE = 41;
literal ISB$K-ST-TY-UNL = 42;
literal ISB$K-ST-TY-FEE = 43;
literal ISB$K-ST-TY-GIN = 44;
literal ISB$K-ST-TY-PIN = 45;
literal ISB$K-ST-TY-MOV = 46;
literal ISB$K-ST-TY-FIN = 47;
literal ISB$K-ST-TY-MIN = 48;
literal ISB$K-ST-TY-RIN = 49;
literal ISB$K-ST-TY-MLI = 50;
```

D 12
15-Sep-1984 23:44:38
15-Sep-1984 23:02:23

VAX-11 Bliss-32 V4.0-742 Page 30
_S255SDUA28:[FORRTL.OBJ]OTSISB.BLF;1 (1)

```

: R0957 0 literal ISB$K-ST-TY-FSE = 51;
: R0958 0 literal ISB$K-ST-TY-MPR = 53;
: R0959 0 literal ISB$K-ST-TY-MRE = 54;
: R0960 0 literal ISB$K-ST-TY-GRFA = 55;
: R0961 0 literal ISB$K-ST-TY-FRFA = 56;
: R0962 0 literal ISB$K-BASSTTYHI = 56;
: R0963 0 literal ISB$K-NEG-LUB = -100;
: R0964 0 literal ISB$K-LEN = 188;
: R0965 0 literal ISB$S-union-1B = 16;
: R0966 0 literal ISB$S-union-2 = 16;
: R0967 0 literal ISB$S-lub-filler = 100;
: R0968 0 FIELD ISB$FIECDSET =
: R0969 0 SET
: R0970 0 ISB$A-RESTARTPC = [-188,0,32,0] ,
: R0971 0 ISB$A-USR-HANDL = [-188,0,32,0] ,
: R0972 0 ISB$S-union-1 = [-188,0,32,0] ,
: R0973 0 ISB$A-MAJ-F-PTR = [-184,0,32,0] ,
: R0974 0 ISB$A-PREVIOUS-LUB = [-184,0,32,0] ,
: R0975 0 ISB$S-union-1A = [-184,0,32,0] ,
: R0976 0 ISB$A-USER-FP = [-180,0,32,0] ,
: R0977 0 ISB$W-FMT-STKP = [-176,0,0,0] ,
: R0978 0 ISB$A-SAVE-PTR = [-176,0,32,0] ,
: R0979 0 ISB$A-SAVE-END = [-172,0,32,0] ,
: R0980 0 ISB$S-structure-1B = [-176,0,0,0] ,
: R0981 0 ISB$S-union-1B = [-176,0,0,0] ,
: R0982 0 ISB$W-FMT-STKR = [-160,0,0,0] ,
: R0983 0 ISB$B-SCA-FAC-D = [-160,0,0,0] ,
: R0984 0 ISB$S-union-2 = [-160,0,0,0] ,
: R0985 0 ISB$B-ERR-NO = [-144,0,8,0] ,
: R0986 0 ISB$B-SCA-FAC = [-144,0,8,1] ,
: R0987 0 ISB$S-union-3 = [-144,0,8,0] ,
: R0988 0 ISB$B-STTM-TYPE = [-143,0,8,0] ,
: R0989 0 ISB$W-FMT-LEN = [-142,0,16,0] ,
: R0990 0 ISB$A-ERR-EQUAL = [-140,0,32,0] ,
: R0991 0 ISB$A-END-EQUAL = [-136,0,32,0] ,
: R0992 0 ISB$A-FMT-BEG = [-132,0,32,0] ,
: R0993 0 ISB$A-FMT-PTR = [-128,0,32,0] ,
: R0994 0 ISB$S-LIS-HEAP-LEN = [-128,0,32,0] ,
: R0995 0 ISB$S-union-3A = [-128,0,32,0] ,
: R0996 0 ISB$A-LIS-STR = [-124,0,32,0] ,
: R0997 0 ISB$B-FMT-P = [-120,0,8,1] ,
: R0998 0 ISB$W-FMT-W = [-119,0,16,0] ,
: R0999 0 ISB$B-FMT-D = [-117,0,8,0] ,
: R1000 0 ISB$B-FMT-E = [-116,0,8,0] ,
: R1001 0 ISB$W-FMT-REP = [-115,0,16,1] ,
: R1002 0 ISB$W-LIS-REP = [-115,0,16,0] ,
: R1003 0 ISB$W-LEN-REM = [-115,0,16,0] ,
: R1004 0 ISB$S-union-4 = [-115,0,16,0] ,
: R1005 0 ISB$B-FMT-CODE = [-113,0,8,0] ,
: R1006 0 ISB$V-FMT-REPRE = [-113,0,1,0] ,
: R1007 0 ISB$R-FMT-CODE-STRUCT = [-113,0,8,0] ,
: R1008 0 ISB$R-FMT-CODE-UNION = [-113,0,8,0] ,
: R1009 0 ISB$B-LIS-CTYPE = [-113,0,8,0] ,
: R1010 0 ISB$S-union-5 = [-113,0,8,0] ,
: R1011 0 ISB$W-FMT-REVER = [-112,0,16,0] ,
: R1012 0 ISB$B-FMT-DEP = [-110,0,8,0] ,
: R1013 0 ISB$W-FMT-FLAGS = [-109,0,16,0] ,

```



```

R1014 0      ISB$B_INP_FLAGS = [-109,0,8,0] ,
R1015 0      ISB$V_BN = [-109,0,1,0] ,
R1016 0      ISB$V_ONLY_E = [-109,1,1,0] ,
R1017 0      ISB$V_ERR_OFLO = [-109,2,1,0] ,
R1018 0      ISB$V_DONTROUND = [-109,3,1,0] ,
R1019 0      ISB$V_SKIPTABS = [-109,4,1,0] ,
R1020 0      ISB$V_EXP_LETTER = [-109,5,1,0] ,
R1021 0      ISB$V_FORCESCALE = [-109,6,1,0] ,
R1022 0      ISB$R_INP_FLAGS_STRUCT = [-109,0,8,0] ,
R1023 0      ISB$R_INP_FLAGS_UNION = [-109,0,8,0] ,
R1024 0      ISB$B_OUT_FLAGS = [-108,0,8,0] ,
R1025 0      ISB$V_SP = [-108,0,1,0] ,
R1026 0      ISB$V_ERR_OFLO = [-108,1,1,0] ,
R1027 0      ISB$R_OUT_FLAGS_STRUCT = [-108,0,8,0] ,
R1028 0      ISB$R_OUT_FLAGS_UNION = [-108,0,8,0] ,
R1029 0      ISB$R_FMT_FLAGS_STRUCT = [-109,0,16,0] ;
R1030 0      ISB$R_FMT_FLAGS_UNION = [-109,0,16,0] ,
R1031 0      ISB$W_STTM_STAT = [-106,0,16,0] ,
R1032 0      ISB$V_P_FORM_CH = [-106,0,2,0] ,
R1033 0      ISB$V_DOLLAR = [-106,2,1,0] ,
R1034 0      ISB$V_USER_ELEM = [-106,3,1,0] ,
R1035 0      ISB$V_SLASH = [-106,4,1,0] ,
R1036 0      ISB$V_LAST_REC = [-106,5,1,0] ,
R1037 0      ISB$V_DE_ENCODE = [-106,6,1,0] ,
R1038 0      ISB$V_LIS_HEAP = [-106,7,1,0] ,
R1039 0      ISB$V_RECURSIVE = [-106,8,1,0] ,
R1040 0      ISB$V_MAT_CONT = [-106,9,1,0] ,
R1041 0      ISB$V_MAT_PRINT = [-106,10,1,0] ,
R1042 0      ISB$V_PRINT_INI = [-106,11,1,0] ,
R1043 0      ISB$V_SINGL_ELEM = [-106,12,1,0] ,
R1044 0      ISB$V_NEED_INIT = [-106,13,1,0] ,
R1045 0      ISB$R_STTM_STAT_STRUCTURE = [-106,0,16,0] ,
R1046 0      ISB$R_STTM_STAT_UNION = [-106,0,16,0] ,
R1047 0      ISB$A_INTFILEND = [-104,0,32,0]
R1048 0      TES:
R1049 0      literal ISB$$ ISB = 189;
R1050 0      MACRO ISB = BLOCK [ISB$$_ISB,byte] FIELD (ISB$FIELDSET) %;

```

F 12
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 32 (1)

: 1051 0 UNDECLARE %QUOTE ISB;
: 1052 0
: 1053 0 REQUIRE 'RTLML:OTSLUB.BLF';

! Logical Unit Block definitions


```

R1054 0  ! *****
R1055 0  ! Created 15-SEP-1984 22:49:15 by VAX-11 SDL V2.0 Source: 15-SEP-1984 22:47:02 _S255$DUA28:[LIBRTL.SRC]OTSL
R1056 0  ! *****
R1057 0
R1058 0
R1059 0  !*** MODULE $LUBDEF ***
R1060 0  literal LUB$K_ORG_SEQUE = 1;
R1061 0  literal LUB$K_ORG_RELAT = 2;
R1062 0  literal LUB$K_ORG_INDEX = 3;
R1063 0  literal LUB$K_ORG_TERMI = 4;
R1064 0  literal LUB$K_ORG_VIRTU = 5;
R1065 0  literal LUB$K_LUN_BPRI = -8;
R1066 0  literal LUB$K_LUN_INPU = -7;
R1067 0  literal LUB$K_LUN_BREAD = -6;
R1068 0  literal LUB$K_LUN_ENCD = -5;
R1069 0  literal LUB$K_LUN_READ = -4;
R1070 0  literal LUB$K_LUN_ACCE = -3;
R1071 0  literal LUB$K_LUN_TYPE = -2;
R1072 0  literal LUB$K_LUN_PRIN = -1;
R1073 0  literal LUB$K_ILUN_MIN = -8;
R1074 0  literal LUB$K_DLUN_MIN = -4;
R1075 0  literal LUB$K_DLUN_MAX = -1;
R1076 0  literal LUB$K_LUN_MIN = 0;
R1077 0  literal LUB$K_LUN_MAX = 119;
R1078 0  literal LUB$K_D_MARGIN = 72;
R1079 0  literal LUB$K_PBUF_SIZ = 80;
R1080 0  literal LUB$K_NEG_BLN = -100;
R1081 0  literal LUB$K_LANG_MIN = 0;
R1082 0  literal LUB$K_LANG_NONE = 0;
R1083 0  literal LUB$K_LANG_BAS = 1;
R1084 0  literal LUB$K_LANG_FOR = 2;
R1085 0  literal LUB$K_LANG_MAX = 2;
R1086 0  literal LUB$K_LUB_CEN = 100;
R1087 0  literal LUB$S_QUEUE = 8;
R1088 0  literal LUB$S_DID = 6;
R1089 0  FIELD LUB$FIECDSET =
R1090 0  SET
R1091 0  LUB$A_UBF = [-100,0,32,0] ,
R1092 0  LUB$W_UNIT_STT3 = [-96,0,16,0] ,
R1093 0  LUB$V_NOECHO = [-96,0,1,0] ,
R1094 0  LUB$V_ONECHR = [-96,1,1,0] ,
R1095 0  LUB$V_CCO = [-96,2,1,0] ,
R1096 0  LUB$V_FIND_LAST = [-96,3,1,0] ,
R1097 0  LUB$V_PTA = [-96,4,1,0] ,
R1098 0  LUB$V_AST_GUARD = [-96,5,1,0] ,
R1099 0  LUB$V_CR = [-96,6,1,0] ,
R1100 0  LUB$V_FTN = [-96,7,1,0] ,
R1101 0  LUB$V_PRN = [-96,8,1,0] ,
R1102 0  LUB$V_NOMARGIN = [-96,9,1,0] ,
R1103 0  LUB$V_USEROPEN = [-96,10,1,0] ,
R1104 0  LUB$V_NOTSEQORG = [-96,11,1,0] ,
R1105 0  LUB$V_ANSI = [-96,12,1,0] ,
R1106 0  LUB$V_RFA_CACHE_ENABLE = [-96,13,1,0] ,
R1107 0  LUB$V_FIECD_USE = [-96,14,1,0] ,
R1108 0  LUB$R_UNIT_STT3_STRUCT = [-96,0,16,0] ,
R1109 0  LUB$R_union_T = [-96,0,16,0] ,
R1110 0  LUB$W_BLS = [-94,0,16,0] ,

```

H 12
15-Sep-1984 23:44:38
15-Sep-1984 23:02:25

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.OBJ]OTSLUB.BLF;1

Page 34
(1)

```
R1111 0 LUBSA_CLOSE = [-92.0,32.0] ,
R1112 0 LUBSQ_QUEUE = [-88.0,0.0]
R1113 0 LUBSA_BUF_PTR = [-80.0,32.0] ,
R1114 0 LUBSA_BUF_END = [-76.0,32.0] ,
R1115 0 LUBSA_BUDDY_PTR = [-72.0,32.0] ,
R1116 0 LUBSA_BUF_BEG = [-68.0,32.0] ,
R1117 0 LUBSA_BUF_HIGH = [-64.0,32.0] ,
R1118 0 LUBSB_ORGAN = [-60.0,8.0] ,
R1119 0 LUBSB_BKS = [-59.0,8.0] ,
R1120 0 LUBSW_LUN = [-58.0,16.0] ,
R1121 0 LUBSL_PRINT_POS = [-56.0,32.0] ,
R1122 0 LUBSA_RFA_CACHE_BEG = [-56.0,32.0] ,
R1123 0 LUBSr_union_1A = [-56.0,32.0] ,
R1124 0 LUBSL_WAIT_TIME = [-52.0,32.0] ,
R1125 0 LUBSA_RFA_CACHE_PTR = [-52.0,32.0] ,
R1126 0 LUBSr_union_1B = [-52.0,32.0] ,
R1127 0 LUBSW_IFI = [-48.0,16.0] ,
R1128 0 LUBSW_RBUF_SIZE = [-46.0,16.0] ,
R1129 0 LUBSW_R_MARGIN = [-44.0,16.0] ,
R1130 0 LUBSW_D_MARGIN = [-42.0,16.0] ,
R1131 0 LUBSB_LANGUAGE = [-40.0,8.0] ,
R1132 0 LUBSB_RFM = [-39.0,8.0] ,
R1133 0 LUBSW_BAS_VFC = [-38.0,16.0] ,
R1134 0 LUBSB_BAS_VFC1 = [-38.0,8.0] ,
R1135 0 LUBSB_BAS_VFC2 = [-37.0,8.0] ,
R1136 0 LUBSR_BAS_VFC_STRUCT = [-38.0,16.0] ,
R1137 0 LUBSr_union_2 = [-38.0,16.0] ,
R1138 0 LUBSA_ASSOC_VAR = [-36.0,32.0] ,
R1139 0 LUBSL_ALQ = [-36.0,32.0] ,
R1140 0 LUBSr_union_3 = [-36.0,32.0] ,
R1141 0 LUBSL_LOG_RECNO = [-32.0,32.0] ,
R1142 0 LUBSL_REC_MAX = [-28.0,32.0] ,
R1143 0 LUBSA_FAB = [-24.0,32.0] ,
R1144 0 LUBSA_RBUF_ADR = [-20.0,32.0] ,
R1145 0 LUBSW_DID = [-16.0,0.0] ,
R1146 0 LUBSB_RAT = [-10.0,8.0] ,
R1147 0 LUBSB_RSL = [-9.0,8.0] ,
R1148 0 LUBSA_RSN = [-8.0,32.0] ,
R1149 0 LUBSW_UNIT_ATTR = [-4.0,16.0] ,
R1150 0 LUBSV_OPENED = [-4.0,1.0] ,
R1151 0 LUBSV_IO_ACTIVE = [-4.1,1.0] ,
R1152 0 LUBSV_READ_ONLY = [-4.2,1.0] ,
R1153 0 LUBSV_OLD_FILE = [-4.3,1.0] ,
R1154 0 LUBSV_DIRECT = [-4.4,1.0] ,
R1155 0 LUBSV_SCRATCH = [-4.5,1.0] ,
R1156 0 LUBSV_DELETE = [-4.6,1.0] ,
R1157 0 LUBSV_PRINT = [-4.7,1.0] ,
R1158 0 LUBSV_FORMATTED = [-4.8,1.0] ,
R1159 0 LUBSV_UNFORMAT = [-4.9,1.0] ,
R1160 0 LUBSV_FIXED = [-4.10,1.0] ,
R1161 0 LUBSV_SEGMENTED = [-4.11,1.0] ,
R1162 0 LUBSV_ASS_VAR_L = [-4.12,1.0] ,
R1163 0 LUBSV_APPEND = [-4.13,1.0] ,
R1164 0 LUBSV_SEQUENTIAL = [-4.14,1.0] ,
R1165 0 LUBSV_KEYED = [-4.15,1.0] ,
R1166 0 LUBSR_UNIT_ATTR_STRUCT = [-4.0,16.0] ,
R1167 0 LUBSr_union_4 = [-4.0,16.0] ,
```


I 12
15-Sep-1984 23:44:38
15-Sep-1984 23:02:25

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[FORRTL.OBJ]OTSLUB.BLF;1 Page 35
(1)

```
: R1168 0      LUB$W_UNIT_STT2 = [-2,0,16,0] ;
: R1169 0      LUB$V_VIRT_RSN = [-2,0,1,0] ;
: R1170 0      LUB$V_ENDFILOPN = [-2,1,1,0] ;
: R1171 0      LUB$V_FORM_CHAR = [-2,2,1,0] ;
: R1172 0      LUB$V_OUTBUF_DR = [-2,3,1,0] ;
: R1173 0      LUB$V_TERM_FOR = [-2,4,1,0] ;
: R1174 0      LUB$V_TERM_DEV = [-2,5,1,0] ;
: R1175 0      LUB$V_FORCIBLE = [-2,6,1,0] ;
: R1176 0      LUB$V_UNIT_0 = [-2,7,1,0] ;
: R1177 0      LUB$V_VA_USE = [-2,8,1,0] ;
: R1178 0      LUB$V_BLK_USE = [-2,9,1,0] ;
: R1179 0      LUB$V_M_STREAM = [-2,10,1,0] ;
: R1180 0      LUB$V_M_STR_C = [-2,11,1,0] ;
: R1181 0      LUB$V_DEALLOC = [-2,12,1,0] ;
: R1182 0      LUB$V_SUBMIT = [-2,13,1,0] ;
: R1183 0      LUB$V_NULLBLNK = [-2,14,1,0] ;
: R1184 0      LUB$V_USER_RBUF = [-2,15,1,0] ;
: R1185 0      LUB$R_UNIT_STT2_STRUCT = [-2,0,16,0] ;
: R1186 0      LUB$r_union_5 = [-2,0,16,0] ;
: R1187 0      TES;
: R1188 0      literal LUB$$LUB = 101;
: R1189 0      MACRO LUB = BLOCK [LUB$$LUB,byte] FIELD (LUB$FIELDSET) %;
```

J 12
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 36
(1)

```
1190 0 UNDECLARE %QUOTE LUB;
1191 0
1192 0
1193 0 !+
1194 0 ! Define macro that declares the CCB structure. Note that it has no
1195 0 ! allocation size - it must be used in a REF declaration.
1196 0 !-
1197 0 MACRO
1198 0 $FOR$CCB_DECL = BLOCK [, BYTE] FIELD (LUB$FIELDSET, ISB$FIELDSET) %;
1199 0
1200 0 REQUIRE 'RTLIN:OTSCCBREQ'; ! OTS CCB data structure definitions
```


R1201 0
R1202 0
R1203 0
R1204 0
R1205 0
R1206 0
R1207 0
R1208 0
R1209 0
R1210 0
R1211 0
R1212 0
R1213 0
R1214 0
R1215 0
R1216 0
R1217 0
R1218 0
R1219 0
R1220 0
R1221 0
R1222 0
R1223 0
R1224 0
R1225 0
R1226 0
R1227 0
R1228 0
R1229 0
R1230 0
R1231 0
R1232 0
R1233 0
R1234 0
R1235 0
R1236 0
R1237 0
R1238 0
R1239 0
R1240 0
R1241 0
R1242 0
R1243 0
R1244 0
R1245 0
R1246 0
R1247 0
R1248 0
R1249 0
R1250 0
R1251 0
R1252 0
R1253 0
R1254 0
R1255 0
R1256 0
R1257 0

+ This file, OTSCCBREQ.REQ, defines the interface to OTSPUSH_CCB Edit: SBL1006
-

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

Revision History:

- 1-001 - Original. JBS 09-JAN-1979
- 1-002 - Change name to OTSCCBREQ.REQ so as not to conflict at system
build time with OTSCCB.B32. SBL 10-May-1979
- 1-003 - Add the definition of the structure for OTS\$AA_LUB_TAB.
JBS 28-JUN-1979
- 1-004 - Add the definition of the field for OTS\$V_LUN_OWNr.
JBS 16-AUG-1979
- 1-005 - Remove PRINT statement, for the new BLISS compiler.
JBS 02-OCT-1979
- 1-006 - Add new structure OTS\$LUN_OWNr_ST used for OTS\$V_LUN_OWNr. This
helps BLISS generate smaller code for references to this structure.

--
+ Define the return codes from OTSPUSH_CCB.
-

LITERAL

OT\$K_PUSH_MIN = 1,	! Smallest valid value
OT\$K_PUSH_OK = 1,	! CCB loaded, I/O not active
OT\$K_PUSH_ACT = 2,	! CCB loaded, I/O active on this LUN
OT\$K_PUSH_FAIL = 3,	! CCB not loaded, out of virtual storage
OT\$K_PUSH_MAX = 3;	! Largest valid value

+ The following structure is used for addressing OTS\$AA_LUB_TAB.

```

R1258 0  ! It is similar to VECTOR, but offsets the index so that certain
R1259 0  ! negative logical unit numbers can be used, and each element is a
R1260 0  ! quadword so as to act as a queue header.
R1261 0  !-
R1262 0
R1263 0  STRUCTURE
R1264 0  OTS$$LUB_TAB_ST [I, SIDE; N, LB, UNIT = 4, EXT = 0] =
R1265 0  [N*UNIT*2]
R1266 0  (OTS$$LUB_TAB_ST + ((SIDE + ((I - LB)*2))*UNIT))<0, 8*UNIT, EXT>;
R1267 0
R1268 0  !+
R1269 0  ! The following structure is similar to BLOCKVECTOR, but allows a low and high
R1270 0  ! bound.
R1271 0  !-
R1272 0
R1273 0  STRUCTURE
R1274 0  OTS$$LUN_OWNR_ST [I, O, P, S, E; L, H, BS, UNIT=1] =
R1275 0  [((H - L) + 1) * BS * UNIT]
R1276 0  (OTS$$LUN_OWNR_ST + (O - L) + ((O + I) * BS * UNIT))<P, S, E>;
R1277 0
R1278 0  !+
R1279 0  ! The following field is used to refer to OTS$$V_LUN_OWNR, which has a bit
R1280 0  ! for each LUB, each block containing a bit for each language.
R1281 0  !-
R1282 0  FIELD
R1283 0  OTS$$V_OWNR_FLD =
R1284 0  SET
R1285 0  OTS$$V_OWNR_BAS = [0, LUB$K_LANG_BAS, 1, 0], ! BASIC
R1286 0  OTS$$V_OWNR_FOR = [0, LUB$K_LANG_FOR, 1, 0], ! FORTRAN
R1287 0  OTS$$V_OWNR = [0, LUB$K_LANG_MIN, (((LUB$K_LANG_MAX-LUB$K_LANG_MIN+%BPUNIT)/%BPUNIT)*%BPUNIT), 0]
R1288 0  TES;
R1289 0  !+
R1290 0  ! The following masks are used to test OTS$$V_OWNR to be sure that only
R1291 0  ! one bit is set.
R1292 0  !-
R1293 0  LITERAL
R1294 0  OTS$$M_OWNR_BAS = 1 ^ LUB$K_LANG_BAS,
R1295 0  OTS$$M_OWNR_FOR = 1 ^ LUB$K_LANG_FOR;
R1296 0
R1297 0  ! End of file OTSCCBREQ.REQ

```


M 12

15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[FORRTL.SRC]FORLIB.REQ;1

Page 39
(1)

: 1298 0
: 1299 0
: 1300 0

REQUIRE 'RTLIN:OTSLNK';

! (Must come after OTSISB and OTSLUB)

! Common linkage definitions

File: OTSLNK.REQ Edit: PLL1035

* This file, OTSLNK.REQ, contains the definitions of all LINKAGE declarations
for BLISS modules

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

```
Author: T. Hastings
1-11 - Add CALL R0. TNH 29-July-78
1-12 - Add JSB_CB_GET. TNH 2-Aug-78
1-13 - Change name to FORLNK.REQ. JBS 14-NOV-78
1-014 - Add copyright notice. JBS 16-NOV-78
1-015 - Change file name to OTSLNK.REQ. JBS 06-DEC-78
1-016 - Add linkage for BMF (Basic major frame ptr) - R11. DGP 17-Dec-78
1-017 - Fix some comments. JBS 18-DEC-78
1-018 - Change JSB_REC0 linkage to save registers needed for CH$FILL. DGP
27-Feb-79
1-019 - Similarly, change JSB_UDF0, since the UDF routines must
preserve the same registers to call the REC routines.
JBS 28-FEB-1979
1-020 - That change causes a similar change in JBS_REC1 and JSB_REC9.
JBS 28-FEB-1979
1-021 - Which in turn causes the same changes in JSB_UDF9.
JBS 28-FEB-1979
1-022 - Which in turn causes the same changes in JBS_DO_READ and
JSB_DO_WRITE. JBS 28-FEB-1979
1-023 - Add linkage JSB_CCB_A1 A0 for PUT relative with count. DGP 02-Mar-79
1-024 - Add linkage JSB_REC_IND for indexed file support. DGP 03-Apr-79
1-025 - Change linkage JSB_REC_IND to take 5 args. DGP 06-Apr-79
1-026 - Add linkage for conversion kernel routine. DGP 27-Jun-79
1-028 - Add linkage for Basic format routines. DGP 30-Jul-79
1-029 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
1-030 - Change JSB_FORMAT A7 to A10. DGP 31-Oct-79
1-031 - Add new linkage JSB_REC_WSL1. DGP 06-Nov-79
```



```

R1358 0 1-032 - Change JSB_UDF0, JSB_FMT0 so that they pass no arguments. SBL 5-Dec-1979
R1359 0 1-033 - Add linkage JSB_FMT1 for Fortran format interpreter. JAW
R1360 0 08-Aug-1981
R1361 0 1-034 - Change JSB_A5_R11 to JSB_A6_R11, JSB_A10_R11 to JSB_A11_R11.
R1362 0 PLL 16-Mar-1982
R1363 0 1-035 - Change JSB_DO_READ, JSB_REC_IND, JSB_REC0. PLL 1-Jun-1982
R1364 0
R1365 0
R1366 0
R1367 0 ++
R1368 0 Linkage definitions for BLISS modules for CALL and JSB routines
R1369 0 The idea is to have all definitions here in one place
R1370 0 so that they can be changed easily and the entire
R1371 0 RIL recompiled.
R1372 0
R1373 0
R1374 0 +
R1375 0 Define symbols for register numbers used to pass parameters from one
R1376 0 module to another. Note: these symbols are used in the modules in
R1377 0 GLOBAL REGISTER declarations rather than below in this REQUIRE file
R1378 0 (where all registers appear as absolute numbers).
R1379 0
R1380 0 LITERAL
R1381 0 K_BMF_REG = 11, ! Register used by Basic compiler to point
R1382 0 ! to last major frame
R1383 0 K_CCB_REG = 11; ! Pointer to LUB/ISB/RAB
R1384 0
R1385 0 +
R1386 0 First define some macros for frequently used combinations.
R1387 0 Do not change the PRESERVE conventions for our sanity.
R1388 0 Also do not change the definitions of these combination, since
R1389 0 they are also used in defining local routines within a module that is CALLED!!!
R1390 0 NOTE: Local routines which are JSBed to from JSB procedures must
R1391 0 have LINKAGE definitions here even though only local procedures.
R1392 0 Otherwise, lose control of NOTUSED registers which must be same
R1393 0 or more inclusive for JSB routines called by JSB routines.
R1394 0 See DO_READ and DO_WRITE for examples.
R1395 0
R1396 0
R1397 0 MACRO
R1398 0
R1399 0 !+
R1400 0 CALL interface with CCB passed in R11 (in and/or out)
R1401 0
R1402 0
R1403 0 CALL_CCB_R11 =
R1404 0 CALL: GLOBAL(CCB=11) %,
R1405 0
R1406 0 !+
R1407 0 JSB interface with CCB passed in R11, 1 arg in R0 and 1 arg in R1 and
R1408 0 only uses R0 and R1
R1409 0
R1410 0
R1411 0 JSB_CCB_A1_A0 =
R1412 0 JSB (REGISTER = 1, REGISTER = 0): GLOBAL(CCB=11) NOTUSED(2,3,4,5,6,7,8,9,10) %,
R1413 0
R1414 0 !+

```

C 13
15-Sep-1984 23:44:38
15-Sep-1984 23:02:30

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[FORRTL.SRC]JOTSLNK.REQ;1 Page 42
(1)

```

: R1415 0      ! CALL interface with BMF passed in R11 (in and/or out)
: R1416 0      !-
: R1417 0
MR1418 0      CALL_BMF_R11 =
: R1419 0      CALL: GLOBAL(BMF=11) %,
: R1420 0
: R1421 0
: R1422 0
: R1423 0      !+
: R1424 0      ! CALL interface with 1st arg in R0
: R1425 0      !-
MR1426 0      CALL_A0 =
: R1427 0      CALL (REGISTER = 0): %,
: R1428 0
: R1429 0
: R1430 0      !+
: R1431 0      ! JSB interface with CCB passed in R11, no args and uses R0-R5
: R1432 0      !-
MR1433 0      JSB_CCB_R5 =
: R1434 0      JSB: GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,
: R1435 0
: R1436 0
: R1437 0      !+
: R1438 0      ! JSB interface with CCB passed in R11, no args and only uses R0, R1
: R1439 0      !-
MR1440 0      JSB_CCB_NO_ARGS =
: R1441 0      JSB: GLOBAL(CCB=11) NOTUSED (2,3,4,5,6,7,8,9,10) %,
: R1442 0
: R1443 0
: R1444 0      !+
: R1445 0      ! JSB interface with CCB passed in R11, 1 arg in R0, and only uses R0, R1
: R1446 0      !-
MR1447 0      JSB_CCB_A0 =
: R1448 0      JSB (REGISTER = 0): GLOBAL(CCB=11) NOTUSED (2,3,4,5,6,7,8,9,10) %,
: R1449 0
: R1450 0
: R1451 0      !+
: R1452 0      ! JSB interface with CCB passed in R11, 1 arg in R0, and preserves
: R1453 0      ! through R5. Needed for MOVC5 or calling routines that use MOVC5.
: R1454 0      !-
MR1455 0      JSB_CCB_A0_R5 =
: R1456 0      JSB (REGISTER = 0): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,
: R1457 0
: R1458 0
: R1459 0      !+
: R1460 0      ! Same as above, but with 2 arguments.
: R1461 0      !-
MR1462 0      JSB_CCB_A1_R5 =
: R1463 0      JSB (REGISTER = 0, REGISTER = 1): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,
: R1464 0
: R1465 0
: R1466 0      !+
: R1467 0      ! JSB interface with CCB passed in R11, 1 arg in R2, and only uses R0, R1, R2
: R1468 0      ! Needed when input arg is referenced after a CALL or JSB,
: R1469 0      ! so do not need to copy to R2.
: R1470 0      !-
: MR1471 0      JSB_CCB_A2 =
```



```

R1472 0      JSB (REGISTER = 2): GLOBAL(CCB=11) NOTUSED (3,4,5,6,7,8,9,10) %,
R1473 0
R1474 0
R1475 0      + JSB interface with CCB passed in R11, 1 arg in R2, and preserves
R1476 0      through R5. Needed in place of JBS_CCB_A2 to do MOVC5, or call
R1477 0      routines which do.
R1478 0      -
R1479 0
MR1480 0      JSB_CCB_A2 R5 =
R1481 0      JSB (REGISTER = 2): GLOBAL(CCB=11) NOTUSED (6,7,8,9,10) %,
R1482 0
R1483 0      +
R1484 0      JSB interface for Fortran format interpreter with CCB passed in
R1485 0      R11, two arguments passed in R10 and R9, and routine value
R1486 0      returned in R8.
R1487 0      -
R1488 0
MR1489 0      JSB_CCB_FMT1 =
MR1490 0      JSB : GLOBAL(CCB = 11, EL_SIZE = 10, DT_SEEN = 9, FMT_CODE = 8)
R1491 0      NOPRESERVE (2,3) NOTUSED (4,5,6,7) %,
R1492 0
R1493 0      +
R1494 0      Support for Indexed files.
R1495 0      Pass arguments (6) in R0:R5 and CCB is passed in R11.
R1496 0      -
R1497 0
MR1498 0      JSB_CCB_A6 R5 =
MR1499 0      JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3,
R1500 0      REGISTER = 4, REGISTER = 5):GLOBAL (CCB = 11) NOTUSED (6, 7, 8, 9, 10) %,
R1501 0
MR1502 0      JSB_CCB_A5 R5 =
MR1503 0      JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3,
R1504 0      REGISTER = 4):GLOBAL (CCB = 11) NOTUSED (6, 7, 8, 9, 10) %,
R1505 0
R1506 0      +
R1507 0      JSB interface with CCB passed in R11, arg1 in R2, arg2 in R0,
R1508 0      and only uses R0, R1, R2.
R1509 0      Needed when input arg1 is referenced after a CALL or JSB, so save
R1510 0      copying to R2.
R1511 0      -
R1512 0
MR1513 0      JSB_CCB_A2 A0 =
R1514 0      JSB (REGISTER = 2, REGISTER = 0): GLOBAL(CCB=11) NOTUSED (3,4,5,6,7,8,9,10) %,
R1515 0
R1516 0      +
R1517 0      JSB interface (no CCB), args in R0 and R9
R1518 0      -
R1519 0
MR1520 0      JSB_A0 A1 R8 =
R1521 0      JSB(REGISTER = 0, REGISTER = 1) : NOPRESERVE (2,3,4,5,6,7,8)%,
R1522 0
R1523 0
R1524 0      +
R1525 0      JSB for Basic format routines - Plain F and E format. Pass
R1526 0      6 args and preserve all other registers. (1 optional arg)
R1527 0      -
R1528 0

```

```
: MR1529 0
: MR1530 0
: R1531 0
: R1532 0
: R1533 0
: R1534 0
: R1535 0
: R1536 0
: R1537 0
MR1538 0
MR1539 0
MR1540 0
: R1541 0
: R1542 0
: R1543 0
: R1544 0
: R1545 0
: R1546 0
: MR1547 0
: R1548 0
: R1549 0

JSB_A6_R11 =
  JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3, REGISTER = 4, REGISTER = 5) :
    PRESERVE (6, 7, 8, 9, 10, 11) %;

!+
JSB for Basic format routines - Fancy F and E formats. Pass 11 args and
!- preserve all other registers. (4 optional args)

JSB_A11_R11 =
  JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2, REGISTER = 3, REGISTER = 4,
    REGISTER = 5, REGISTER = 6, REGISTER = 7, REGISTER = 8, REGISTER = 9, REGISTER = 10) :
    PRESERVE (11) %;

!+
JSB interface (without CCB), no args in registers
!-

JSB_NO_ARGS =
  JSB: NOTUSED (2,3,4,5,6,7,8,9,10) %;
```



```
R1550 0
R1551 0
R1552 0
R1553 0
R1554 0
R1555 0
R1556 0
R1557 0
R1558 0
R1559 0
R1560 0
R1561 0
R1562 0
R1563 0
R1564 0
R1565 0
R1566 0
R1567 0
R1568 0
R1569 0
R1570 0
R1571 0
R1572 0
R1573 0
R1574 0
R1575 0
R1576 0
R1577 0
R1578 0
R1579 0
R1580 0
R1581 0
R1582 0
R1583 0
R1584 0
R1585 0
R1586 0
R1587 0
R1588 0
R1589 0
R1590 0
R1591 0
R1592 0
R1593 0
R1594 0
R1595 0
R1596 0
R1597 0
R1598 0
R1599 0
R1600 0
R1601 0
R1602 0
R1603 0
R1604 0
R1605 0
R1606 0
```

```
++
Now define the LINKAGE declarations.
Use names associated with the entry point rather than
the type of linkage, so that we can easily change
the linkage for an entry point without changing that
for other entry points.
Note: entry points that are dispatched to using a table
must have the same linkage name. In this case the LINKAGE
name is associated with the name of the dispatch table and
the call is made using the general LINKAGE form.

NOTUSED restriction!!! Because each JSB declaration must be
aware of all JSB routines which are in turn called. The NOTUSED
registers can only be the same as the caller (if also a JSB routine)
or include additional registers as well as being the same.
Thus, this file documents the calling tree for JSB linkages
so that the NOTUSED declarations can be kept in agreement.
--

LINKAGE

+
Default CALL using CCB as a GLOBAL register,
all args in arg list.
-

CALL_CCB =          CALL_CCB_R11,

+
CALL from BASIC compiled code, which uses R11 to point to the
major frame.
-

CALL_BMF =          CALL_BMF_R11,

+
This is a linkage for BAS$$REC_WSL1 to allow one arg to be passed.
-

JSB_REC_WSL1 =      JSB_CCB_A0_R5,

+
CALL passing first arg in R0.
Used by FORENTRY module to make multiple entry points
all branch to FIOBEG.
-

CALL_FIOBEG =       CALL_A0,

+
UDF initialization (user data formatting level of abstraction)
Arg is adr. of format statement.
JSBs to record level initialization (JSB_REC0).
-

JSB_UDF0 =          JSB_CCB_R5,

+

```

```

R1607 0      | JSB to plain formatting routines for Basic.
R1608 0      | -
R1609 0      |
R1610 0      | JSB_FORMAT_A6 = JSB_A6_R11,
R1611 0      |
R1612 0      | +
R1613 0      | JSB to fancy formatting routines for Basic.
R1614 0      | -
R1615 0      |
R1616 0      | JSB_FORMAT_A11 = JSB_A11_R11,
R1617 0      |
R1618 0      | +
R1619 0      | UDF termination (user data formatting level of abstraction)
R1620 0      | JSBs to DO_READ (JSB_DO_READ) or DO_WRITE (JSB_DO_WRITE).
R1621 0      | -
R1622 0      |
R1623 0      | JSB_UDF9 =          JSB_CCB_R5,
R1624 0      |
R1625 0      | +
R1626 0      | UDF read routine
R1627 0      | JSBs to record level (JSB_REC1).
R1628 0      | -
R1629 0      |
R1630 0      | JSB_DO_READ =       JSB_CCB_A1_R5,
R1631 0      |
R1632 0      | +
R1633 0      | UDF write routine
R1634 0      | JSBs to record level (JSB_REC1).
R1635 0      | -
R1636 0      |
R1637 0      | JSB_DO_WRITE =      JSB_CCB_A0_R5,
R1638 0      |
R1639 0      | +
R1640 0      | Format interpreter initialization: FORMAT_ADR = arg is adr. of format statement
R1641 0      | JSBs to nothing.
R1642 0      | -
R1643 0      |
R1644 0      | JSB_FMT0 =          JSB_CCB_NO_ARGS,
R1645 0      |
R1646 0      | +
R1647 0      | Format interpreter main processing
R1648 0      | -
R1649 0      |
R1650 0      | JSB_FMT1 =          JSB_CCB_FMT1,
R1651 0      |
R1652 0      | +
R1653 0      | JSB to REC level of index file support
R1654 0      | -
R1655 0      |
R1656 0      | JSB_REC_IND = JSB_CCB_A5_R5,
R1657 0      | JSB_REC_IND1 = JSB_CCB_A6_R5,
R1658 0      |
R1659 0      | +
R1660 0      | Record level RMS interface level initialization.
R1661 0      | JSBs to nothing.
R1662 0      | -
R1663 0

```


H 13
15-Sep-1984 23:44:38
15-Sep-1984 23:02:30

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]OTSLNK.REQ;1 Page 47
(2)

```
: R1664 0 JSB_REC0 = JSB_CCB_R5,
: R1665 0 JSB_REC2 = JSB_CCB_A0_R5,
: R1666 0
: R1667 0
: R1668 0 | + Record level RMS interface level finished one buffer
: R1669 0 | JSBs to nothing.
: R1670 0 | -
: R1671 0
: R1672 0 JSB_REC1 = JSB_CCB_R5,
: R1673 0
: R1674 0 | +
: R1675 0 | Record level RMS interface termination of statement.
: R1676 0 | JSBs to nothing.
: R1677 0 | -
: R1678 0
: R1679 0 JSB_REC9 = JSB_CCB_R5,
: R1680 0
: R1681 0 | +
: R1682 0 | Push current LUB/ISB/RAB: LOGICAL_UNIT is unit no., LUN_MIN is min. no.
: R1683 0 | JSBs to nothing.
: R1684 0 | -
: R1685 0
: R1686 0 JSB_CB_PUSH = JSB_CCB_A2_A0,
: R1687 0
: R1688 0 | +
: R1689 0 | PUT relative with count
: R1690 0 | -
: R1691 0
: R1692 0 JSB_PUT = JSB_CCB_A1_A0,
: R1693 0
: R1694 0 | +
: R1695 0 | Pop current LUB/ISB/RAB
: R1696 0 | JSBs to nothing.
: R1697 0 | -
: R1698 0
: R1699 0 JSB_CB_POP = JSB_CCB_NO_ARGS,
: R1700 0
: R1701 0 | +
: R1702 0 | Return current LUB/ISB/RAB to free storage (open error or close)
: R1703 0 | JSBs to nothing.
: R1704 0 | -
: R1705 0
: R1706 0 JSB_CB_RET = JSB_CCB_NO_ARGS,
: R1707 0
: R1708 0 | +
: R1709 0 | Get adr. of current LIB/ISB/RAB (called only from non-shared routines)
: R1710 0 | since harder to have a data entry vector which is the same
: R1711 0 | if module were to become shared or vice versa.
: R1712 0 | -
: R1713 0
: R1714 0 JSB_CB_GET = JSB_CCB_NO_ARGS,
: R1715 0
: R1716 0 | +
: R1717 0 | JSB to kernel conversion routine
: R1718 0 | -
: R1719 0
: R1720 0 JSB_CVT_KERNEL = JSB_A0_A1_R8,
```

I 13
15-Sep-1984 23:44:38
15-Sep-1984 23:02:30

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]OTSLNK.REQ;1 Page 48
(2)

```
: R1721 0
: R1722 0
: R1723 0
: R1724 0
: R1725 0
: R1726 0
: R1727 0
: R1728 0
!      JSB to CALL_VFE routine, args on stack
      JSB_CALL_VFE =      JSB_NO_ARGS;
      End of file OTSLNK.REQ
```


J 13
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_S255SDUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 49
(1)

: 1729 0
: 1730 0

REQUIRE 'RTLIN:OTSMAC';

! Common macros

R1731 0
R1732 0
R1733 0
R1734 0
R1735 0
R1736 0
R1737 0
R1738 0
R1739 0
R1740 0
R1741 0
R1742 0
R1743 0
R1744 0
R1745 0
R1746 0
R1747 0
R1748 0
R1749 0
R1750 0
R1751 0
R1752 0
R1753 0
R1754 0
R1755 0
R1756 0
R1757 0
R1758 0
R1759 0
R1760 0
R1761 0
R1762 0
R1763 0
R1764 0
R1765 0
R1766 0
R1767 0
R1768 0
R1769 0
R1770 0
R1771 0
R1772 0
R1773 0
R1774 0
R1775 0
R1776 0
R1777 0
R1778 0
R1779 0
R1780 0
R1781 0
R1782 0
R1783 0
R1784 0
R1785 0
R1786 0
R1787 0

* This file, OTSMAC.REQ, defines OTS macros.
Edit: SBL1039

*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*
* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*
* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*
* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*

Author: T. Hastings

- 1-25 - REQUIRE LPSECT. TNH 19-Dec-77
- 1-26 - Remove SET CB BASE(). JMT 12-Apr-78
- 1-27 - Use RTLIN: logical name in REQUIRE. TNH 28-Apr-78
- 1-28 - Define ADR_VECTOR. TNH 7-June-78
- 1-30 - Change name to FORMAC.REQ (with apologies to Dick Gruen)
and change name of LPSECT to RTLPSECT JBS 14-NOV-78
- 1-031 - Add a copyright notice JBS 16-NOV-78
- 1-032 - Change file name to OTSMAC.REQ and remove REQUIRE of RTLPSECT.
(Let users of OTSMAC.REQ also REQUIRE RTLPSECT.) JBS 06-DEC-78
- 1-033 - Add offsets and lengths of the dispatch tables. JBS 25-JUN-1979
- 1-034 - Make them weak globals so they can be used by macro routines.
JBS 26-JUN-1979
- 1-035 - Remove FORTRAN offsets and lengths (moved to ISB). JBS for SBL
12-JUL-1979
- 1-036 - Remove BASIC offsets and lengths (moved to ISB). JBS 12-JUL-1979
- 1-037 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
- 1-038 - Add COPY_BYTE_A, COPY_WORD_A, COPY_LONG_A, COPY_QUAD_A
macros. SBL 18-Dec-1979
- 1-039 - Add ONE_OF macro. SBL 18-Dec-1981

*
Macro for writing a character string and then advancing pointer
Designed so that it is placed on the left side of a substitution
statement. Anticipates feature being added to BLISS as a form
for CH\$WCHAR_A (DESPA) if looks good.


```

R1788 0
R1789 0
R1790 0
R1791 0
R1792 0
MR1793 0
MR1794 0
MR1795 0
MR1796 0
MR1797 0
R1798 0
R1799 0
R1800 0
R1801 0
R1802 0
R1803 0
R1804 0
R1805 0
R1806 0
R1807 0
R1808 0
R1809 0
R1810 0
MR1811 0
MR1812 0
R1813 0
R1814 0
R1815 0
R1816 0
R1817 0
R1818 0
R1819 0
R1820 0
R1821 0
R1822 0
R1823 0
MR1824 0
R1825 0
R1826 0
R1827 0
R1828 0
R1829 0
R1830 0
R1831 0
R1832 0
R1833 0
R1834 0
R1835 0
R1836 0
R1837 0
R1838 0
R1839 0
R1840 0
R1841 0
R1842 0
PR1843 0
PR1844 0

! Call: CH_WCHAR_A (CS_POINTER_ADR.ma.r) = ... ;
!
MACRO
    CH_WCHAR_A (CS_POINTER_ADR) =
        (LOCAL T;
         T = .CS_POINTER_ADR;
         CS_POINTER_ADR = CH$PLUS (.CS_POINTER_ADR, 1);
         .T<0,8> %;

!+
! Macro for writing a character without advancing the pointer.
! Desinged so that is placed on the left of a substitution statement.
! Anticipates feature being added to BLISS as a form
! for CH$WCHAR (DSTPV) if looks good.
!
! Call: CH_WCHAR (CS_POINTER.ra.v) = ... ;
!
MACRO
    CH_WCHAR (CS_POINTER_VAL) =
        (CS_POINTER_VAL)<0,8> %;

!+
! Macros for processing the compiled format text byte strings.
!
MACRO
    RBYTE_A(P) = (P = .P+1; .(.P-1)<0, 8>) %,
    RWORD_A(P) = (P = .P+2; .(.P-2)<0,16>) %,
    RLONG_A(P) = (P = .P+4; .(.P-4)<0,32>) %,

    CALL VFE(P)=
        T (LOCAL T; T = .(.P)<0,32>; P = .P+4; .T+.P) ( ) ) %;

!+
! Macros for copying values referenced by pointers.
!
MACRO
    COPY_BYTE_A (S,D) = (D=.D+1; (.D-1)<0,8>=RBYTE_A(S)) %,
    COPY_WORD_A (S,D) = (D=.D+2; (.D-2)<0,16>=RWORD_A(S)) %,
    COPY_LONG_A (S,D) = (D=.D+4; (.D-4)<0,32>=RLONG_A(S)) %,
    COPY_QUAD_A (S,D) = ((.D)<0,32>=.(.S)<0,32>; (.D+4)<0,32>=.(.S+4)<0,32>; D=.D+8; S=.S+8) %;

!+
! Macro to complete the transportable character pointer notion.
! Everywhere that an address (A) can be specified in BLISS,
! allow a character pointer with mnemonic P (rather than CP to keep one letter)
!
%BLISS32 (
    MACRO

```

M 13
15-Sep-1984 23:44:38
15-Sep-1984 23:02:27

VAX-11 Bliss-32 V4.0-742
_S255\$DUA28:[FORRTL.SRC]JOTSMAC.REQ;1 Page 52
(1)

PR1845 0
PR1846 0
PR1847 0
PR1848 0
PR1849 0
PR1850 0
PR1851 0
R1852 0
R1853 0
R1854 0
R1855 0
R1856 0
R1857 0
R1858 0
MR1859 0
R1860 0
R1861 0
R1862 0
R1863 0
R1864 0
R1865 0
R1866 0
R1867 0
R1868 0
R1869 0
R1870 0
R1871 0
R1872 0
R1873 0
R1874 0
R1875 0

LSSP = LSSA %,
LEQP = LEQA %,
EQLP = EQLA %,
NEQP = NEQA %,
GEQP = GEQA %,
GTRP = GTRA %,
MAXP = MAXA %,
MINP = MINA %);

+
:- Clear a vector of BLISS values (transportable)
:-

MACRO
FILL VAL (VALUE, LENGTH, ADDRESS) =
%BLISS32 (CH\$FILL (VALUE, (LENGTH) * %UPVAL, ADDRESS)) %;

+
:- Allocate string descriptor
Rest of descriptor symbols are defined in SRMDEF.MDL
But currently no way in MDL to define a macro

To declare and allocate a descriptor:

LOCAL
name: DSC\$DESCRIPTOR;

MACRO
DSC\$DESCRIPTOR = BLOCK[8, BYTE] %; ! MDL requires BYTE

THE "ONE OF" MACRO

MACRO

Macros to determine if the value of an expression is one of a set of specified small-integer values. These macros can be used only if the following conditions are met:

The value to be tested is in the range 0 through 127.

The values to be tested for are all in the range 0 through 31.

Example:

IF ONE_OF (.X, 1,3,5) ...

The code generated is much more efficient than a series of comparisons (provided that the values being tested are all compile-time constants).

```
XBMSK [A]=
  %IF NOT %CTCE(A) %THEN %ERRORMACRO('ONE_OF argument not a CTCE') %FI
  %IF (A GTRU 31) %THEN %ERRORMACRO('ONE_OF constant greater than 31') %FI
  (1 ^ (31 - (A))) %,
```

```
BMSK []=
  0 OR XBMSK_(%REMAINING)) %,
```

```
XCMP [A,B,C]=
  %IF %LENGTH EQL 3
  %THEN
    ((A EQLU B) OR (A EQLU C))
  %ELSE
    (A EQLU B)
  %FI %,
```

```
ONE_OF(A)=
  %IF %LENGTH LEQ 1 %THEN %ERRORMACRO('Too few arguments to ONE_OF') %FI
  %IF %LENGTH LEQ 3
  %THEN
    XCMP_(A,%REMAINING)
  %ELSE
    (( ( BMSK_(%REMAINING) ) ^ (A)) LSS 0)
  %FI %;
```

End of file OTSMAC.REQ

B 14
15-Sep-1984 23:44:38
15-Sep-1984 22:44:59

VAX-11 Bliss-32 V4.0-742
_\$255\$DUA28:[FORRTL.SRC]FORLIB.REQ;1 Page 54
(1)

: 1924 0
: 1925 0 ! End of file FORLIB.REQ

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	5	0	581	00:01.0

COMMAND QUALIFIERS

: BLISS/LIBRARY=LIB\$:/LIST=LIS\$:/SOURCE=REQUIRE SRC\$:FORLIB

: Run Time: 00:19.9
: Elapsed Time: 01:19.3
: Lines/CPU Min: 5818
: Lexemes/CPU-Min: 26644
: Memory Used: 164 pages
: Library Precompilation Complete

0181 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

FORINTUND
LIS

FORIOBEG
LIS

FORIOEND
LIS

FORLEX
LIS

FORMSG
LIS

FORMLTAB
LIS

FORINQUIR
LIS

FORIOELEM
LIS

FORIODATE
LIS

FORLIB
LIS